DESCRIPTION OF THE COURSE

Name of the course: **PARALLEL PROGRAMMING**	Code: **BCSEe33**	Semester: **5**
Type of teaching: **Lectures, Laboratory Work, CW / CP**	Hours per week: **L – 2 hours, LW – 2 hours**	Number of credits: **5**

LECTURER:
Prof. PhD Ognyan Nakov (FCSC), email: nakov@tu-sofia.bg, TU-Sofia
Prof. PhD Daniela Goceva (FCSC), email: dgoceva@tu-sofia.bg, TU-Sofia
Assoc. Prof PhD Antonia Tasheva (FCSC), email: atasheva@tu-sofia.bg, TU-Sofia
Assoc. Prof PhD Iva Nikolova (FCSC), email: inni@tu-sofia.bg, TU-Sofia
Assist. Prof PhD Yavor Tomov (FCSC), email: yavor_tomov@tu-sofia.bg, TU-Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for the students of specialty Computer science and Engineering in the bachelor programme of the Faculty of Computer Systems and Technologies.

AIMS AND OBJECTIVES OF THE COURSE: At the end of the course the students are expected to know the concepts, principles, models and paradigms of parallel information processing and to apply them in the design and development of parallel software; to be able to develop effective programming implementations, to make profiles, estimate and analyze the performance, to verify and evaluate the parallel software implementations.

DESCRIPTIONOFTHECOURSE: Fundamental principles of parallel programming. Styles of parallel programming. Synthesis of parallel algorithms. Basic approaches and stages of synthesis. Development of parallel software applications. Parallel computing models. Implementing parallelism in applications. Principles of vectorization. Problems of automatic and explicit vectorization. Parallel programming for memory platforms based on shared memory. Implementation of functional and data parallelism. APIs for multithreaded programming. Parallel programming for computing platforms based on distributed memory. Message passing model. Parallel programming with combining multithreading and multiprocessor processing. Parallel Programming for Graphic Processors. Concurrency in the C ++ standard. Introduction to Asynchronous Tasks. Comparison with multi-threaded processing. Introduction to asynchronous control after .NET version 4.5. Asynchronous input / output processing (web and mobile technologies). Asynchronous file input / output across platforms. Asynchronous and background processing. Asynchronous desktop and web applications. Asynchronous UI, WPF Dispatcher. Asynchronous MVC processing. Comparison with asynchronous Web API operations. Building Asynchronous Mobile Applications. Process control libraries, threads and tasks. Working with services. Multi-threaded Java. Performing competitive phase tasks. Exchange of data between competing tasks. Finishing and attaching tasks asynchronously. Synchronization of multi-threaded applications in Java.Analysis and evaluation of parallel performance. Measurement approaches and metrics to assess performance. Scalability of parallel programs.

PREREQUISITES: Programming languages, Programming environments, High-performance computer systems.

TYPE OF TEACHING: Lectures using video-presentation with beamer, laboratory works end with presentation of the results, parallelism profiles and estimation of the performance parameters of the parallel system for the certain task.

METHOD OF ASSESSMENT: Exam during the exam session two academic hours in the form of a test with open and closed questions and problems or tasks related to writing parallel programming code (80%), semester assignments (20%).

LANGUAGES OF INSTRUCTION: Bulgarian.

BIBLIOGRAPHY:

1. Презентации на лекциите по паралелно програмиране в платформата за еОбучение „Moodle“ (cs.tu-sofia.bg)
2. Parallel Programming: for Multicore and Cluster Systems, Thomas Rauber, Gudula Rünger, ISBN: 978-3642378003 (2013)
3. Structured Parallel Programming, Michael McCool, James Reinders, Arch Robison, Publisher: Morgan Kaufmann (2012)
4. An Introduction to General-Purpose GPU Programming, Jason Sanders, Edward Kandrot, ISBN: 978-0131387683, (2012)
5. Introduction to Concurrency in Programming Languages, Matthew J. Sottile, Timothy G. Mattson, Craig E. Rasmussen, Chapman and Hall/CRC (2009).
6. Java concurrency in practice, Goetz, Brian, and Tim Peierls.. Pearson Education, (2006).
7. Patterns for Parallel Programming, Timothy G. Mattson, Beverly A. Sanders, Berna L. Massingill, Addison-Wesley, (2005)
8. The Art of Multiprocessor Programming, Maurice Herlihy, Nir Shavit, ISBN: 978-0123973375, (2012)
9. Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming, Stephen Cleary, "O'Reilly Media, Inc.", May 15, 2014
10. Pro Asynchronous Programming with .NET, Richard Blewett, Andrew Clymer, Apress, Dec 18, 2013
11. Asynchronous Android Programming, Helder Vasconcelos, Packt Publishing Ltd, Jul 29, 2016

DESCRIPTION OF THE COURSE

Name of the course **Advanced Software Technologies (C#)**	Code: **BCSCe34**	Semester: **5**
Type of teaching: Lectures and laboratory work	Lessons per week: L – 2 hours; LW – 2 hours	Number of credits: **5**

LECTURER:

Assoc. Prof. Antoniya Tasheva, PhD, MSc, Eng., TU - Sofia, Faculty of Computer Systems and Technologies, Department of Computer Systems (CS), e-mail: atasheva@tu-sofia.bg

COURSE STATUS IN THE CURRICULUM: Compulsory course for bachelor degree students of the specialty ”Computer Science and Engineering” of the Faculty of Computer Systems and Technologies at the Technical University of Sofia.

AIMS AND OBJECTIVES OF THE COURSE:

To develop a student’s ability to solve problems using advanced software technologies of the .NET Framework, .NET Core and the programming language C# separately and in teamwork. The subjects studied and the skills developed in this course are applied in the development of the diploma projects.

DESCRIPTION OF THE COURSE:

The main topics concern: Object-oriented programming methodologies, Managed languages; introduction to C# programming language: classes, methods, properties; inheritance, interfaces, polymorphism, indexers, attributes, exceptions, delegates and events; Advanced C#: complex types, type manipulation, Reflection, Generics, LINQ, Encryption techniques. What’s new in C# 7.X. Capabilities of .NET Core 2.0

PREREQUISITES:

The course assumes that the students are aware of the basic operation of a computer system, programming fundamentals and object-oriented programming.

TEACHING METHODS:

Lectures, using a beamer, case studies, laboratory work in teams, preparation of exercises and laboratory works, solve a problem using C#.

METHOD OF ASSESSMENT: Exam.

LANGUAGE OF INSTRUCTION: English.

RECOMMENDED LITERATURE:

1. Lecture notes, http://cs.tu-sofia.bg/
2. Sander Rossel, Object-Oriented Programming in C# Succinctly, Syncfusion, 2016
3. Joe Mayo, C# Succinctly, Syncfusion, 2015
4. Troelsen, A., Japikse, Ph., Pro C# 7 With .NET and .NET Core, Apress, 2017
5. Microsoft, What's new in C#, https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/
6. Svetlin Nakov , Veselin Kolev et al , Introduction to Programming with C #, Telerik Academy for Software Engineers , 2011.

DESCRIPTION OF THE COURSE

Name of the course **Integrated Digital Technologies**	Code: **BCSCe35**	Semester: 8
Type of teaching: Lectures and Seminars	Lessons per week: L – 2 hours; S – 1 hour	Number of credits: **5**

LECTURER:

Assoc. Prof. PhD Georgi Angelov (FCST) – tel.: 965 2570, email: angelov@ecad.tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory course for the students of specialty Computer and Software Engineering in the bachelor programme of the Faculty of Computer Systems and Technologies.

AIMS AND OBJECTIVES OF THE COURSE: The aim of the course is the students to have basic knowledge of the design and manufacturing of VLSIs and of the main technology processes of production of VLSI and VLSI with mixed structures, of the Hybrid Integrated Circuits and Multichip Modules, as well Electronics system, based on them, also knowledge of the Electro-Magnetic Compatibility, Noises and Electromigration in the VLSIs and Electronic Systems and use these knowledge in solving of engineering problems.

DESCRIPTION OF THE COURSE: The main topics concern: Design and Manufacturing of VLSIs and VLSIs with mixed structure; Main technology processes of production of modern monolithic submicron VLSI; Technology processes of Hybrid Integrated Circuits and Multichip Modules; Interconnections; SMD Technologies; Electro-Magnetic Compatibility, Noises and Electromigration in the VLSIs and Electronic Systems.

PREREQUISITES: Physics, Digital Circuits, Analysis and Synthesis of Logical Circuits.

TEACHING METHODS: Lectures using video-presentation with beamer. Seminar using presentation of SMD and Wave Soldering equipment, visual control equipment. Visits in enterprises are planned.

METHOD OF ASSESSMENT: Two one-hour assessments at mid and end of semester (total 70%), laboratories (20%), Lecture attendance - (10%)

LANGUAGE OF INSTRUCTION: English.

BIBLIOGRAPHY: 1. Weste, Neil, Harris, David, *Principles of CMOS VLSI Design - A Circuits and Systems Perspective, 4th Edition* Addison-Wesley, MA, 2010. 2. Ali, Jamnia, *Practical Guide to the Packaging of Electronics, Second Edition: Thermal and Mechanical Design and Analysis*, 2013. 3. К.Фильов, Т.Таков, Съвременни методи за тестване на СГИС, ТУ-София, 2008. 4. Фирмена литература на Intel, HIC от 2012 и 2013г. (включена в лекциите)

DESCRIPTION OF THE COURSE

Name of the course **Software Design and Software Testing**	Code: **BCSCe36**	Semester: **5**
Type of teaching: Lectures and Laboratory work	Lessons per week: L – 2 hours, LW – 2 hours	Number of credits: **5**

LECTURER:
Assoc. Prof. PhD Adelina Aleksieva-Petrova (FCST), tel.: 965 26 52, email: aaleksieva@tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for the students speciality "Computer Science and Engineering" – bachelor degree (Faculty of Computer Systems and Technologies of TU-Sofia).

AIMS AND OBJECTIVES OF THE COURSE: The aim of the course is to teach the students with the practice concepts of software engineering – design, implementation, debugging and maintenance of software products.

DESCRIPTION OF THE COURSE: The course includes problems concerning models and stages of software life cycle - software development method including design, development, testing and debugging, support and maintenance. Various programming styles are under discussion – imperative (procedure oriented) using C,C++,C# as implementation language, logic programming using Prolog and functional programming using Lisp.

PREREQUISITES: Basic knowledge, practical skills and experience in universal procedure oriented programming languages like C, C++, Java. Material covered is to be useful for software engineers – designers, managers, developers and quality assurance engineers.

TEACHING METHODS: Lectures (with slides, multimedia projector) and additional text materials; web site of the course; laboratory work (based on instructions) with a tutorial for every laboratory theme.

METHOD OF ASSESSMENT: Written examination (test work for fixed time).

INSTRUCTION LANGUAGE: English

BIBLIOGRAPHY:
Bonev S., Technology of Programming, Sofia, Siela, 2000 (in Bulgarian); Schach St., Object-Oriented and Classical Software Engineering, Asken Assoc. Inc. Publ., IRWIN, 8th ed., 2010.
B. Hambling (editor), Software Testing ISEB Foundation, BCS Publishing Products, 2009.
A. Hunt, D. Thomas, Pragmatic Unit Testing in C# with NUnit, The Pragmatic Bookshelf, 3e, 2010, Sommerville I., Software Engineering, Addison Wesley, 9e, 2010, K.Louden, Programming Languages, Principles and Practice, IE Cengage Learning, 3e, 2012.
R. Sebesta, Concepts of Programming Languages, Addison Wesley, 10e, 2012.

DESCRIPTION OF THE COURSE

Name of the course **Embedded System**	Code: **BCSCe37**	Semester: **5**
Type of teaching: Lectures and laboratory work	Lessons per week: L – 2 hours; LW – 2 hours	Number of credits: **4**

LECTURER:
Asist.Prof.PhD Kamelia Raynova, (FCST), tel.: 965 2164, e-mail: kkaneva@tu-sofia.bg
Technical University of Sofia, Bulgaria

COURSE STATUS IN THE CURRICULUM: Compulsory course for the students of specialty Computer Science and Engineering in the bachelor program of the Faculty of Computer Systems and Technologies.

AIMS AND OBJECTIVES OF THE COURSE: The aim of the course is to help students to learn and to be able to apply the suitable methods, approaches, and technical tools for analysis, design, and application of popular microprocessor families, ACISC, and single-chip microcomputers in accordance with their engineering needs. Also, the students should have the skills to upgrade themselves their knowledge in this broad engineering area.

DESCRIPTION OF THE COURSE: Course examines: the requirements for "embedded systems", algorithm design, the design specifics of the input and output interface software systems for design of embedded systems design the properties of a processor, dual-and hierarchical architectures of embedded systems, tools and methods setup and documentation of embedded systems.

PREREQUISITES: The successful completion of this course is based on the knowledge and practical skills, obtained in some previous courses as: „Theoretical electrotechnics", „Semiconductor devices", „Eelectrical measurements".

TEACHING METHODS: Lectures, using slides, case studies, laboratory and course work, work in teams and course work description preparation. Students have at their disposal these lecture notes well in advance, so they are able to prepare themselves for further fruitful discussions. Various supported materials are also available at: http://cs-tusofia.eu/ incl. labs preparation instructions, keywords, references, etc

METHOD OF ASSESSMENT: Exam during the exam session with duration of two academic hours (90 min), students give written answers to set of 7 questions. Each correct answer gives 0,5 or 1 point (n). Mark "excellent" (6) is awarded when $n \geq 5,5$; "very well" (5) - when $n \geq 4,5$; "well" (4) – in case of $n \geq 3,5$, "satisfactory" (3) - when n=3. The final mark is arranged as a result from the written exam's results (80%) and this of the laboratory exercises (20%).

INSTRUCTION LANGUAGE: English

BIBLIOGRAPHY:

1. Лекционни записки, презентационни слайдове (lecture notes).
2. EMBEDDED HARDWARE know it all. Newnes
3. EMBEDDED SYSTEMS WORLD CLASS DESIGNS. Newnes
4. Steve Heath, Embedded Systems Design, Second edition, 2003
5. Tammy Noergaard, Embedded Systems Architecture, Third edition, 2005
6. Neil Weste, Kamran Eshraghian, Principles of CMOS VLSI Design, Addison-Wesley Publishing, Second edition, 2001
7. Stuart R. Ball, Analog Interfacing to Embedded Microprocessor Systems, Second edition, 2004

DESCRIPTION OF THE COURSE

Name of the course **Programming Frameworks**	Code: **BCSCe36**	Semester: **6**
Type of teaching: Lectures and laboratory work	Lessons per week: L – 2 hours; LW – 2 hour	Number of credits: **5**

LECTURER:
Prof. PhD Ognian Nakov Nakov (FCST), tel.: 965 3513, email: nakov@tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for students speciality "Computer science and engineering" – bachelor degree (Faculty of Computer Systems and Technologies of TU-Sofia).

AIMS AND OBJECTIVES OF THE COURSE: The course introduces traditional and modern means for Windows programming in.NET Frameworks; graphical interface (GDI/GDI+) programming; document/view architecture; multitasking basics and memory management; object construction/destruction and usability in classical/visual/.NET environments; fundamentals of serialization; internet programming basics – API functions, basic classes and class hierarchies; exception handling mechanisms, writing code complete and code prone to hacker attacks.

DESCRIPTION OF THE COURSE: The course introduces specific and common language adjustments in connection with the development environment. Principles and realization of inter-language connections in program modules of a specific product. Traditional and modern means for Windows programming, including.NET Frameworks. Graphical interface programming (GDI / GDI+). Document/view architecture. Multitasking basics. Memory management, objects construction/ destruction and lifetime. Generations. Serialization. Internet programming basics – API functions basic classes and class hierarchies. Exception handling mechanisms and writing code complete. Code, prone to hackers attacks.

PREREQUISITES: Basic knowledge in operation systems, universal program languages, software engineering, such as and knowledge about special features, structure and functionality of computer devices and system.

TEACHING METHODS: Lectures in multimedia variant; developed web site with all lecture and practical matherials of the course; practical work in laboratory. Published tutorial for every laboratory theme.

METHOD OF ASSESSMENT: Written examination with developed individual program.

INSTRUCTION LANGUAGE: Bulgarian.

BIBLIOGRAPHY:

1.Наков О. и колектив, ПРОГРАМНИ СРЕДИ - .NET, ръководство за лабораторни упражнения – част 2, издателство на ТУ- София , 2004.
2. John Sharp, Microsoft Visual C# 2013 Step by Step (Step by Step Developer), 2013.
3.Templeman Julian, Microsoft Visual C++/CLI Step by Step (Step by Step Developer), Microsoft Pres, 2013.
4.Bruce Johnson, Professional Visual Studio 2013 (Wrox Programmer to Programmer), Wrox, 2014.
Допълнителна литература
5.Prata, S. C++ Primer Plus (6th Edition), Addison-Wesley Professional, 2011.
6.Stroustrup, B. Programming: Principles and Practice Using C++ (2nd Edition). Addison-Wesley Professional, 2014

DESCRIPTION OF THE COURSE

Name of the course **Agent-Based Technologies**	Code: **BCSCe40**	Semester: 6
Type of teaching: Lectures and laboratory work	Lessons per week: L – 2 hours; LW – 1 hour	Number of credits: **4**

LECTURERS:
Assoc. Prof. PhD Adelina Aleksieva-Petrova, phone.: 965 26 52, email: aaleksieva@tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for the students specialty Computer science and engineering BEng programme of the Faculty of Computer Systems and Technologies

AIMS AND OBJECTIVES OF THE COURSE: At the end of the course the students are expected to be able to apply the methodologies and software tools for design and deployment of agent based applications, to assess in which cases use agent-based applications and how to implement the communication between them, to implement intelligent agents, multi-agent systems and mobile agents and use them in solving of engineering problems.

DESCRIPTION OF THE COURSE: The main topics concern: *Agent based software technologies* – motivation and application of agent based technologies in Web business services; *Web services and software agents* – interoperability, integration of software agents; *Agents* - definition, infrastructure, interaction of agents; *Service oriented architecture based on agents*; *Intelligent agent* - definition, properties, characteristics, classification, tools for deployment; *Multi-agent systems* - nature, basic elements, properties, communication and distribution of the task; *GAIA methodology for designing of multi-agent architecture* - roles model and interaction model, analysis and design; *Methodology based on messages for agent-based analysis and design - concepts and notations, analysis; Mobile agents - nature, field of application; Java-based agent platforms* - specifications IEEE FIPA, architecture of FIPA, platforms Cougaar, AgentFactory, 3APL platform, Jason (AgentSpeak APL); *JADE platform* - architecture, packages, messaging service, administration and development of applications, programming, communication, detection of agents; *UBIWARE platform* - architecture, programming language for semantic agents (S-APL).

PREREQUISITES: Programming and Computing III, Programming languages, Analysis and design of algorithms.

TEACHING METHODS: Lectures, using slides, case studies, laboratory, work in teams.

METHOD OF ASSESSMENT: Two academic hour assessments. The exam consists of answers on questions and tasks.

INSTRUCTION LANGUAGE: Bulgarian.

BIBLIOGRAPHY: 1. Lectures and laboratory exercises http://cs.tu-sofia.bg. 2. Aleksieva-Petrova A., Gancheva V., Manual in Agent-Based Technologies, Technical University of Sofia, 2012. 3. Bellifemine, F.L., Caire, G., Greenwood, D. Developing Multi-Agent Systems with JADE, John Willey & Sons Ltd, 2007. 4. Wooldridge, M. J. An Introduction to MultiAgent Systems, John Willey & Sons Ltd, 2009. 5. Danny Weyns, Architecture-Based Design of Multi-Agent Systems, Springer, ISBN 3642010636, 2010. 6. Mark d'Inverno, Michael Luck, Understanding Agent Systems (Springer Series on Agent Technology), Springer, ISBN-10: 3642073824, 2010. 7. Gerhard Weiss, Multiagent Systems (Intelligent Robotics and Autonomous Agents series), The MIT Press, ISBN-10: 0262018896, 2013.

DESCRIPTION OF THE COURSE

Name of the course **Computer Graphics**	Code: **BCSCe41**	Semester: **6**
Type of teaching: Lectures and Laboratory work, Course project	Lessons per week: L – 2 hours, LW – 2 hours	Number of credits: **5**

LECTURER:

Prof. PhD Milena Lazarova (FCST), tel. 965-3285, email: milaz@tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory course for the "bachelor" degree students in specialty "Computer science and engineering" in Faculty "Computer systems and technologies", TU-Sofia.

AIMS AND OBJECTIVES OF THE COURSE: To provide an overview of modern interactive graphics systems and an insight into the most important special techniques required for developing efficient interactive graphics applications. The students should obtain an understanding of the important special tools required and problems encountered while building graphics applications.

DESCRIPTION OF THE COURSE: The main topics concern: overview of modern interactive graphics systems; 2D model transformations using homogeneous coordinates; 2D viewing pipeline – graphics windows and viewports; basic raster algorithms; developing efficient interactive graphics applications; 3D image modeling; projections; 3D viewing pipeline; elimination of hidden surfaces; illumination models; color models; image rendering using local and global illumination models; ray tracing; spline curves and surfaces; fractals; computer art and animation. OpenGL is used as the platform for practical C programming exercises, and as an example of a modern system which incorporates many of the fundamental ideas of computer graphics.

PREREQUISITES: Mathematics, Algorithms & Data Structures; Programming Languages and Programming Environments.

TEACHING METHODS: Lectures, using slides, case studies, demo programs and multimedia presentations, course works assignments

METHOD OF ASSESSMENT: final exam.

INSTRUCTION LANGUAGE: Bulgarian.

BIBLIOGRAPHY: Course web site: cs.tu-sofia.bg g/bgmoodle/course /view.php?id=71; Hughes J., A. van Dam, M. McGuire, D. Sklar, J. Foley, S. Feiner, K. Akeley, Computer Graphics: Principles and Practice, Addison-Wesley, 2013; Shreiner D., The Khronos OpenGL ARB Working Group, B. Licea-Kane, G. Sellers, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Addison-Wesley, 8th ed., 2013; Wolff D., OpenGL 4.0 Shading Language Cookbook, Packt Publishing, 2011; Shirley P., M. Ashikhmin, S. Marschner, Fundamentals of Computer Graphics, AK Peters, 2009; Gortler S., Foundations of 3D Computer Graphics, MIT Press, 2012; Angel E., D. Shreiner, Interactive Computer Graphics: A Top-Down Approach with Shader-Based, Addison-Wesley, 2011; Wright R., N. Haemel, G. Sellers, B. Lipchak, OpenGL SuperBible: Comprehensive Tutorial and Reference, Addison-Wesley, 2011; Matsuda K., R. Lea, WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL, Addison-Wesley, 2013.

DESCRIPTION OF THE COURSE

Name of the course: **Bioinformatics**	Code: **BCSCe42**	Semester: **6**
Type of teaching: Lectures, Laboratory, Work Course work	Hours per week: L – 2 hours, LW – 2 hours	Number of credits: **5**

LECTURER:

Assoc. Prof. PhD Dilyana Budakova (FCST) – tel.:, email: dilyana_budakova@yahoo.com
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for the students of specialty Computer Science and Engineering in the bachelor programme of the Faculty of Computer Systems and Technologies.

AIMS AND OBJECTIVES OF THE COURSE: At the end of the course the students are expected to know the biological data and techniques for their detection and extraction; know the algorithms and methods for storing and analyzing biological data and to apply them in practice; acquire knowledge and skills to compare and to use various software tools for analysis, modeling, processing and visualization of nucleotide and protein sequences.

DESCRIPTION OF THE COURSE: The main topics concern: Fundamentals of bioinformatics. Biological sequences. Analysis of biological sequences. Basic principles of molecular biology. Biological databases. Protein and DNA / RNA data bases. Sequences alignment. Arrange the pairs of sequences. Methods and algorithms for sequence alignment. Software tools for bioinformatics. Prediction of the RNA secondary structure - Methods and software tools. Evolutionary analysis of biological data. Software to predict genes.

PREREQUISITES: Object-oriented programming, parallel programming, high-performance computer systems, data structures and algorithms.

TYPE OF TEACHING: Lectures using video-presentation with beamer, laboratory works end with presentation of the results for the certain task.

METHOD OF ASSESSMENT: Exam during the exam session with duration two academic hours, students give written answers to 3 compulsory and 5 optional questions, problems or tasks (60%), laboratory works (25%), course work (15%).

LANGUAGES OF INSTRUCTION: English.

BIBLIOGRAPHY:

1. E-learning materials - lectures and laboratory exercises http://cs.tu-sofia.bg/ Bioinformatics.
2. Borovska P., Gancheva V., Georgiev I., Parallel Algorithms and models for in silico biological experiments on high-performance computing clusters and supercomputer BlueGene / P, TU-Sofia, 2012.
3. **Heitor S. Lopes, Leonardo M. Cruz**, *Computational Biology and Applied Bioinformatics*, on-line book, **ISBN-13**: 9789533076294, pp. 442, 2011
4. **David Mount,** Bioinformatics: Sequence and Genome Analysis, ISBN-13: 978-0879697129, book, March 2013
5. **Mitchell L. Model,** Bioinformatics Programming Using Python: Practical Programming for Biological Data, ISBN-13: 978-0596154509, book, 2013
6. **Sung W.K**., *Algorithms in Bioinformatics: A Practical Introduction*, CRC Press, 2009.
7. **Ramsden J.**, *Bioinformatics: An Introduction (Computational Biology*), Springer, 2010.

DESCRIPTION OF THE COURSE

Name of the course **Java Technologies**	Code: **BCSCe43**	Semester: **6**
Type of teaching: **Lectures and laboratory work**	Lessons per week: **L – 2 hours; LW – 2 hour**	Number of credits: **5**

LECTURER:

Prof. Ph.D. Ivan Momtchev(FCST) – tel.: 965 2052, email: ivan.momtchev@tu-sofia.bg

Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for the specialty Computer Science and Engineering Bachelor Eng program.

AIMS AND OBJECTIVES OF THE COURSE: At the end of the course, participants will be able to use the techniques and tools provided by the programming language Java, to apply the exception handling mechanism, object serialization, to master multithreads programming approaches and techniques.

DESCRIPTION OF THE COURSE: Revision of Java Programming Techniques - Classes and Objects; Controlling Access to Members of a Class; Collections in Java – Arrays, Lists; Strings; Console Input; Inheritance, Polymorphism; Interfaces; AWT and Swing introduction.

Exceptions handling – safety, checked and unchecked. Java I/O -Output Streams, input Streams, Filter Streams, Readers and Writers, Threads – Creating, States, Running, Synchronization, Deadlock, Thread Scheduling, Thread Pools.

PREREQUISITES:

Basic skills in object-oriented programming.

TEACHING METHODS:

Lectures, using internet sites and multimedia projector, case studies, laboratory and course work, work in teams, protocols and course work description preparation and defense

METHOD OF ASSESSMENT:

Two - hours assessments at end of semester, or course work preparation and defense.

INSTRUCTION LANGUAGE: English

BIBLIOGRAPHY

1. Ivan Momtchev, Java Technologies, http://refg.tu-sofia.bg/JTech, 2007- 2013

2. Ivan Momtchev, http://refg.tu-sofia.bg/JavaBg 2004 - 2013

3. Ivan Momtchev, http://refg.tu-sofia.bg/AdvJava/; 2011-2013

4. The Java Tutorial http://download.oracle.com/javase/tutorial/ 2013

5. Bruce Eckel , Thinking in Java (4th Edition), Prentice Hall 2008;

DESCRIPTION OF THE COURSE

Name of the course **Programming Technologies for Secure Code**	Code: **BCSCe44**	Semester: 6
Type of teaching: Lectures and laboratory work	Lessons per week: L – 2 hours; LW – 2 hour	Number of credits: **5**

LECTURER:
Prof. Ognian Nakov Nakov Ph.D. (FCST), – tel.: 965 3513, email: nakov@tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for students speciality "Computer science and engineering" – bachelor degree (Faculty of Computer Systems and Technologies of TU-Sofia).

AIMS AND OBJECTIVES OF THE COURSE: The course introduces the vulnerability of the code in different programming technologies local or external attacks, as well as software tools and practices for protection. The aim of the discipline is provide knowledge about information protection and to give students skills for identification of possible risks in certain systems and for application of proper means for protection.

DESCRIPTION OF THE COURSE: The course includes: threat modeling, the principle of least privilege, defense in depth, authentication, luring attack, running as non-privileged user, writing code that can be used by a non-admin, auditing, security context, security context in the .NET Framework, token, privileges, granting end revoking privileges, daemon, run a program as another user, impersonation, impersonate a user, impersonation in ASP.NET, COM authentication level, COM impersonation, initialize security for COM, configurating security for a COM client, store secrets on a computer, programmatically log off or reboot the computer, overruns attacks**,** safe exception handling, security enhancements in the .NET Framework, points to cryptography in .NET, ASP security, server-side security controls, defining roles, configuration file encryption, cryptographic elements, protecting secret data, using PKCS #5 to make the attacker's job harder, Protecting secrets in Windows 2000 and later, difference between LSA secrets and DPAPI, encrypting secret data in memory, different ways of storing secret data – rising the security bar up.

PREREQUISITES: Basic knowledge in operation systems, universal program languages, software engineering, such as and knowledge about special features, structure and functionality of computer devices and system.

TEACHING METHODS: Lectures in multimedia variant; developed web site with all lecture and practical matherials of the course; practical work in laboratory. Published tutorial for every laboratory theme.

METHOD OF ASSESSMENT: Written examination with developed individual program.

INSTRUCTION LANGUAGE: English.

BIBLIOGRAPHY: 1. Хауърд, М. Д. Лебланк, Писане на сигурен код, СофтПрес, 2004. 2. Hamid R. Nemati and Li Yang, Applied Cryptography for Cyber Security and Defense: Information Encryption and Cyphering, IGI Global, 2010. 3. Bryan Sullivan and Vincent Liu, Web Application Security, A Beginner's Guide, McGraw-Hill Osborne Media, 2011. 4. Michal Zalewski, The Tangled Web: A Guide to Securing Modern Web Applications, No Starch Press, 2011. 5. Mike Shema, Hacking Web Apps: Detecting and Preventing Web Application Security Problems, Syngress, 2012. 6. Michael Howard, David LeBlanc and John Viega, 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them, McGraw-Hill Osborne Media, 2009.

DESCRIPTION OF THE COURSE

Name of the course: **Systems On Chip Design**	Code: **BCSCe45.2**	Semester: **7**
Type of teaching: Lectures and laboratory work	Lessons per week: L – 2 hours; LW – 2 hour	Number of credits: 4

LECTURER:

Assoc. Prof. Ph.D. Peter Manoilov (FCST), tel.: 0895 590 576, email : p.manoilov@mail.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Optional course for regular education of the student specialty "*COMPUTER SCIENCE AND ENGINEERING* ", Bachelor degree, Faculty of Computer Systems and Technologies, Technical University-Sofia.

AIMS AND OBJECTIVES OF THE COURSE The aim of this course is to give knowledge and skills in the area of CAD design of Systems on Chip (SoC) on currently usable VLSI chips.

DESCRIPTION OF THE COURSE: The syllabus considers the basic topics of the theory and practice of Systems on Chip (SoC) design : microelectronic technology, art of circuit design, computer architectures, hardware description languages (HDL), design methodology and CAD systems for hardware and software co-design on contemporary VLSI chips.

PREREQUISITES: Basic knowledge on microelectronics, digital and mixed signal circuit design, computer architectures, programming languages.

TEACHING METHODS: Lectures, using white board and slide presentations, laboratory work, using PC-based CAD systems and programmable VLSI chips on reference boards.

METHOD OF ASSESSMENT: Written exam (70%) and laboratory work (30%).

BIBLIOGRAPHY:

1. Manoilov P. – Lectures on System on Chip Design, in Moodle, TU – Sofia, 2013.

2. Chu P. – FPGA prototyping by VHDL examples, John Wiley & Sons, 2008.

3 . Mano M., M. Ciletti – Digital Design with an introduction to the Verilog HDL, Prentice Hall, 2013.

4. Wolf W. - Modern VLSI Design, IP – based design, Prentice Hall, 2009.

5. Stefanov T., E. Deprettere , Marinov M., Nikolov H., Popov A. – Embedded Systems, TU – Sofia, 2012.

.

DESCRIPTION OF THE COURSE

Name of the course: **Embedded system programming**	Code: **BCSCe45.1**	Semester: **6**
Type of teaching: Lectures, Laboratory work	Lessons per week: L – 2 hours, LW – 1 hours	Number of credits: **4**

LECTURER:
Assoc. Prof. Ph.D. D. Gotseva (FKSU) – tel.: 965 2338, email: dgoceva@tu-sofia.bg
Technical University of Sofia

COURSE STATUS IN THE CURRICULUM: Compulsory for the students' specialty "Computer Science and Engineering" of the Faculty of Computer Systems and Control of Technical University of Sofia – bachelor degree.

AIMS AND OBJECTIVES OF THE COURSE: The aim of this course is to provide the basis for a education in the fundamentals of embedded systems technology and to show the way in which this field is currently developing and is likely to develop in the future.

DESCRIPTION OF THE COURSE: The main topics concern: Role of the embedded systems in modern life; Classification and characteristics of the modern embedded system; Systems with standart architecture; Systems with a single processor unit architecture; types of embedded systems for creating projects; Working with a system in real life; Realtime working models; Structure of an embedded system mechanism; I/O interfaces – classification; Analogue i/o peripherals; Protocols and drivers for programming input/output; communication protocols in the IoT devices; connecting outside devices – examples; Software and programming methods; Arduino; Raspberry Pi; NodeMCU; Intel Edison; STM8; Assembly language; programming with C, C++, Java, Python, Lua, HTML, CSS, javaScript etc..

PREREQUISITES: Digital Schematics, Microprocessor tehnologies, Programming Languages

TEACHING METHODS: Lectures, using slides, case studies, laboratory and course work, work in teams with models and embedded system prototypes.

METHOD OF ASSESSMENT: Written exam (60%), laboratory works (40%).

INSTRUCTION LANGUAGE: Bulgarian

BIBLIOGRAPHY:
1. http : //www.msdnaa. net/curri culum/pfu . aspx?5 96 I 2. https://www.arduino.cc/ 3. https://www.raspberr)'pi.org/ 4. http://nodemcu.readthedocs.io/en/latest/ 5. http://www.instructables.com/ 6. Иванов И.Е., О. Маринов, Микропроцесорна техника 1 част, ТУ София 2008 7. Liu, J., Real Time Systems, Prentice Hall, 2000 8. Heath S., Embedded Systems Design, s.e. , Newnes, Elsevier, London, 2003 9. Ganssle J. at al., Embedded Hardware, Newnes, Elsevier, London, 2008 10. Labrosse J. at al., Embedded Software, Newnes, Elsevier, London, 2008 11. Laplante Ph., Real-Time Systems - Design and Analysis, 3'd ed., IEEE Press, 2004