South Ural State University
Faculty of Computational Mathematics and Informatics

Course Descriptions in Fundamental Computer Science and IT
MAJOR: Technologies for Game Development

		ECTS cr
B.1.02	Mathematical Foundation of Information Security	3
B.1.03	Algorithmic Foundation of Multimedia Technologies	2
B.1.05	Java Programming	3
V.1.01	Mobile Programming	4
DV.1.01.01	Markup Languages	3
DV.1.02.01	Advanced Methods of Software Development	2
B.2.01	Information Technology Analysis	2
B.2.02	Object-oriented CASE Technologies	2
B.2.03	Object Databases	2
B.2.04	Distributed Object Technologies	3
B.2.05	Distributed and Parallel Programming	4
V.2.01	Seminar on Advanced Technologies for Computer Game Development	4
V.2.02	Artificial Intelligence for Games	2
V.2.03	Computer Animation and Simulation	2
DV.2.01.01	Game Platform Development	3
DV.2.02.01	Social Network Game Development	2

B.1.02	***MATHEMATICAL FOUNDATIONS OF INFORMATION SECURITY*** ***3 ECTS cr***
Year and Semester	Year 1 Semester 1
Teacher(s)	Rifkhat Aleev, Doctor of Science, Professor of System Programming Department.
Aims	The student obtains basic skills in mathematical methods of information security. Upon completion of the course, the student will be able to implement basic algorithms of information protection.
Content	Factorization of large numbers. Discrete logarithm. Groups. Rings, fields. Basics of information theory. Linear codes. Error detection and correction. Symmetric and asymmetric ciphers. Diffie-Hellman requirements. RSA cryptosystem*.* Digital signature. Computer system, access, security policy. Identification and authentication. Password-based protection systems. Clark-Wilson model.
Modes of Study	Lectures 18 h. Practical assignments 36 h. Self-study 54 h. Total 108 h.
Evaluation	2-5
Study Materials	Materials are delivered/announced during classes.
Prerequisites	Bachelor courses are required: B.2.04 Algebra and Geometry B.2.04 Finite graph theory and its applications B.03.02 Discrete mathematics

B.1.03	***ALGORITHMIC FOUNDATION OF MULTIMEDIA TECHNOLOGIES*** ***2 ECTS cr***
Year and Semester	Year 2 Semester 3
Teacher(s)	Mikhail Mezhenin, Master of Science, Assistant Lecturer of System Programming Department
Aims	The student obtains basic knowledge of algorithms used to encode, compress and process multimedia data. Upon completion of the course, the student will be able to design and implement algorithms and applications for working with different multimedia data.
Content	Modern multimedia technologies. Data encoding and compression: run-length encoding, working with binary data. Image processing: Netpbm project, encoding and converting full-color and grayscale images, dithering, Floyd-Steinberg algorithm. Multimedia libraries: FFmpeg, Simple DirectMedia Layer. Media-player development: reading, demuxing, decoding and playing multimedia data.
Modes of Study	Practical assignments 36 h. Self-study 36 h. Total 72 h.
Evaluation	Passed|Failed. Credit test – 30%, practical assignments – 70%.
Study Materials	Materials are delivered/announced during classes.

B.1.05	***JAVA PROGRAMMING*** ***3 ECTS cr***
Year and Semester	Year 1 Semester 1
Teacher(s)	Artem Nabirkin, Lecturer of System Programming Department
Aims	The student obtains basic skills in Java programming language. Upon completion of the course, the student will able to develop high-quality Java applications using modern design techniques (OOP, design patterns, etc.).
Content	Introduction to the Java language. Java programming environment. Data types and type conversion. Objects, classes, packages. Object oriented programming in Java basics. Operators and the structure of the code. Exception handling and debugging. Collections. Execution of threads, synchronization, work with files. java.lang, java.awt packages. Swing library, user interface development. The garbage collector. Basic design patterns. Internationalization.
Modes of Study	Practical assignments 54 h. Self-study 54 h. Total 108 h.
Evaluation	2-5. Exam 50 %, practical assignments 50 %.
Study Materials	Materials are delivered/announced during classes.

V.1.01	***MOBILE PROGRAMMING*** ***4 ECTS cr***
Year and Semester	Year 2 Semester 3
Teacher(s)	Aleksandr Gorskih, Master of Science, Assistant Lecturer of System Programming Department
Aims	The student obtains basic skills in mobile programming. Upon completion of the course, the student will be able to design and implement applications for mobile devices.
Content	Introduction: xCode, Objective-C, Cocoa API. Mobile GUI development: StoryBoard, segue, gesture recognition, AnimationKit, IBAction, IBOutlet. Data processing in iOS: iCloud, CoreData, MapKit, accounts framework, accelerate framework, CoreBluetooth, CoreLocation. Game development: OpenGL ES 2.0, AV Foundation, Game Center, GameKit. iOS application development framework: iOS MVC, OCMock, OCUnit, CI (Continuous Integration).
Modes of Study	Practical assignments 54 h. Self-study 54 h. Total 108 h.
Evaluation	2-5. Practical assignments 50 %, exam 50 %.
Study Materials	Materials are delivered/announced during classes.

DV.1.01.01	***MARKUP LANGUAGES*** ***3 ECTS cr***
Year and Semester	Year 1 Semester 1
Teacher(s)	Elena Ivanova, Master of Science, Senior Lecturer of System Programming Department
Aims	The student obtains basic skills in markup languages. Upon completion of the course, the student will be able to apply World Wide Web Consortium (W3C) technologies in document processing.

Content	Introduction to markup languages: motivation, classification and basic elements – tags, elements and attributes. Hypertext Markup Language (HTML). Cascading Style Sheets (CSS). XML technologies. Document Type Definition (DTD). Navigating in XML-documents using XPath language. Transformation and visualization of XML-documents using XSL (eXtensible Stylesheet Language). XML Schema. Linking of XML-elements using XLink and XPointer languages. Scalable Vector Graphics (SVG) language.
Modes of Study	Practical assignments 54 h. Self-study 54 h. Total 108 h
Evaluation	2-5. Practical assignments 50 %, exam 50 %.
Study Materials	Materials are delivered/announced during classes.
Prerequisites	Influences Web-based programming course (bachelor).

DV.1.02.01	***ADVANCED METHODS OF SOFTWARE DEVELOPMENT*** ***2 ECTS cr***
Year and Semester	Year 1 Semester 2
Teacher(s)	Olga Ivanova, Candidate of Science, Associate Professor of System Programming Department
Aims	The student obtains basic skills in object-oriented methods for information systems development. Upon completion of the course, the student will be able to design and implement applications using design patterns, test-driven development, refactoring and SOLID methodology.
Content	General principles of object-oriented design. The concept of clean code. The SOLID methodology. Test-driven development (TDD) and refactoring. Basic design patterns: Abstract Factory, Singleton, Adapter, Bridge, etc. MVC (Model-View-Controller) patterns. Basic templates for design of enterprise applications: Allocator, Plug-in, Selected interface, etc. ORM technology and examples of its implementation.
Modes of Study	Practical assignments 36 h. Course project (self-study) 36 h. Total 72 h
Evaluation	Passed|Failed. Credit test 20%, practical assignments 40%, course project 40%.
Study Materials	Materials are delivered/announced during classes.
Prerequisites	Object-oriented CASE technologies

B.2.01	***INFORMATION TECHNOLOGY ANALYSIS*** ***2 ECTS cr***
Year and Semester	Year 2 Semester 3
Teacher(s)	Fedianina Raisa, Senior lecturer of System Programming Department
Aims	The student obtains basic skills in IT standards and global information infrastructure technologies. Upon completion of the course, the student will be able to develop profiles of information systems and perform conformance testing of such profiles.
Content	The concept of open systems; system of IT standards and its organizational structure. Profiles of open systems environment (OSE profiles). Methodology and system of POSIX OSE standards. OSI System

	of standards. Specification of network protocols and their services. Methodology and technology of OSI conformance testing. Concept of global information infrastructure.
Modes of Study	Practical assignments 36 h. Self-study 36 h. Total 72 h.
Evaluation	2-5. Practical assignments 50 %, exam 50 %.
Study Materials	Materials are delivered/announced during classes.
Prerequisites	Object-oriented CASE-technologies

B.2.02	***OBJECT-ORIENTED CASE TECHNOLOGIES 2 ECTS cr***
Year and Semester	Year 1 Semester 1
Teacher(s)	Olga Ivanova, Candidate of Science, Associate Professor of System Programming Department
Aims	The student obtains basic skills in information systems design using UML. Upon completion of the course, the student will be able to apply the UML-based modeling tools and engineering methods for the software design and implementation.
Content	Analysis and Extraction of Classes. The Class Diagram. Diagrams of the Internal Structure, Components and Accommodation. Use Case Diagram. The Interaction Diagram. The State Diagram. The Activity Diagram.
Modes of Study	Practical assignments 36 h Course project 33 h Credit test 3 h Total 72 h
Evaluation	Passed|Failed. Credit test 30%, practical assignments 70%.
Study Materials	Materials are delivered/announced during classes.

B.2.03	***OBJECT DATABASES 2 ECTS cr***
Year and Semester	Year 2 Semester 3
Teacher(s)	Mikhail Zymbler, Candidate of Science, Associate Professor of System Programming Department
Aims	The student obtains basic skills in database systems based on object model. Upon completion of the course, the student will be able to design and implement applications for object-oriented, object-relational, XML, graph, document-oriented and geospatial databases.
Content	Motivation of Object databases: impedance mismatch problem, manifests in database technologies, Object Database Management Group (ODMG) and its activities. Object-relational databases: column objects, row objects, nested tables, subtypes and supertypes (Oracle DBMS as an example). Object-oriented databases: ODMG architecture, ODL (Object Definition Language), OQL (Object Query Language), OML (Object Manipulation Language). XML databases and XQuery language (Sedna XML DBMS as an example). Document-oriented databases (MongoDB DBMS as an example). Graph databases (Neo4j DBMS as an example). Geospatial databases (PostGIS DBMS as an example).
Modes of Study	Practical assignments 36 h. Self-study 36 h. Total 72 h.

Evaluation	Passed|Failed. Credit test 30%, practical assignments 70%.
Study Materials	Materials are delivered/announced during classes.

B.2.04	***DISTRIBUTED OBJECT TECHNOLOGIES 3 ECTS cr***
Year and Semester	Year 1 Semester 2
Teacher(s)	Gleb Radchenko, Candidate of Science, Associate Professor of the System Programming Department. Dmitry Nenazhenko, Master of Science, Assistant Lecturer of the System Programming Department.
Aims	The student obtains basic skills in distributed computing systems and service-oriented architectures. Upon completion of the course, the student will be able to design and implement distributed applications based on RMI, web-services and cloud computing approach.
Content	Definition, classification and history of Distributed Computing Systems. The CAP theorem. RMI and distributed object technologies middleware stacks: RPC, Java RMI, .NET Remoting, CORBA. Service Oriented Architecture: definition, basic concepts, good practices. Basic standards of XML Web Services (WSDL, SOAP, WS-Security, WS-Addressing). The concept of REST Services. Principles and technology of peer-to-peer systems. The concept of Grid. Grid platforms: UNICORE. Cloud computing technologies and platforms: Windows Azure, Amazon EC2, Google Cloud Platform. Mass computing systems: BOINC platform.
Modes of Study	Practical assignments 36 h. Lectures 18 h. Self-study 54 h. Total 108 h.
Evaluation	Passed|Failed. Credit test 30%, practical assignments 70%.
Study Materials	1) Robert Daigneau. Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services. 2011. 352 p. 2) Kai Hwang, Jack Dongarra, Geoffrey C. Fox. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Morgan Kaufmann, 2011. 672 p. 3) John Rhoton, Risto Haukioja. Cloud Computing Architected: Solution Design Handbook. Recursive Press, 2011. 385 p. 4) David Patterson, Armando Fox. Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing. Strawberry Canyon LLC, 2012. 412 p. 5)Tomas Erl. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, 2005. 792 p. Additional materials are delivered/announced during classes.
Prerequisites	Students should be able to develop cross-platform software on high-level language (Java). Students should know the principles of object-oriented software design.

B.2.05	***DISTRIBUTED AND PARALLEL PROGRAMMING 4 ECTS cr***
Year and Semester	Year 1, 2 Semester 2, 3

Teacher(s)	Tatyana Lymar, Candidate of Science, Associate Professor of System Programming Department
Aims	The student obtains basic skills in parallel programming. Upon completion of the course, the student will be able to design and implement parallel algorithms and applications for multi-core, multiprocessor and distributed computing systems.
Content	Goals and objectives of parallel processing. Types of parallel processing. Architectures of parallel computing systems. Methods for evaluating the performance of multiprocessor systems. Principles for the development of parallel algorithms. Technological development cycle: partitioning, communication, agglomeration and mapping. Complexity analysis of parallel algorithms. Speedup and efficiency of parallel algorithms. Parallel programming for multiprocessor systems with distributed memory, MPI standard. Parallel programming for multiprocessor systems with shared memory, OpenMP standard.
Modes of Study	Lectures 18 h Practical assignments 54 h. Self-study 72 h Total 144 h
Evaluation	2-5. Exam test 50%, practical assignments 50%.
Study Materials	Materials are delivered/announced during classes.

V.2.01	***SEMINAR ON ADVANCED TECHNOLOGIES FOR COMPUTER GAME DEVELOPMENT 4 ECTS cr***
Year and Semester	Year 1, 2 Semester 2, 3
Teacher(s)	Pavel Verman, Master of Science, Assistant of System Programming Department
Aims	The student obtains basic practical skills in team computer game development.
Content	Game design document, gameplay proposal, draft project, technical project. Game prototype development. Alpha-version of the game. Beta-version of the game. Game testing. Final version of the game. Demo-version of the game. Presentation of the game. Technical support and maintenance.
Modes of Study	Practical assignments 72 h. Self-study 72 h. Total 144 h.
Evaluation	Passed|Failed. Practical assignments 100%.
Study Materials	Materials are delivered/announced during classes.

V.2.02	***ARTIFICIAL INTELLIGENCE FOR GAMES 2 ECTS cr***
Year and Semester	Year 1 Semester 1
Teacher(s)	Ruslan Miniakhmetov, Master of Science, Lecturer of System Programming Department
Aims	The student obtains basic skills in AI algorithms development and using common AI paradigms for game development through examples of famous algorithms and its usage in typical game situations. Upon completion of the course, the student will be able to apply methods and algorithms of artificial intelligence in games development to create

	exciting, high quality and efficient computer games.
Content	Model graphic game development. Algorithms for chasing, evasion, shooting and collision detection. The A* and genetic path-finding algorithms. Decision-making weight-based and clustering-based algorithms. Optimization of the brute-force algorithm by means of Alpha-Beta pruning. Implementation of first-order logic for an intelligent agent.
Modes of Study	Practical assignments 36 h. Self-study 36 h. Total 72 h.
Evaluation	Passed|Failed. Credit test 30%, practical assignments 70%.
Study Materials	Materials are delivered/announced during classes.

V.2.03	***COMPUTER ANIMATION AND SIMULATION 2 ECTS cr***
Year and Semester	Year 1 Semester 2
Teacher(s)	Pavel Kostenetskiy, Candidate of Science, Associate Professor of System Programming Department Gorskih Alexander, Master of Science, Assistant of System Programming Department
Aims	The student obtains basic skills in computer animation and simulation. Upon completion of the course, the student will be able to work with shader development frameworks, advanced 3D modeling and physics simulation software.
Content	Computer animation and simulation technologies: shader programming, 3D modeling, illumination models, procedural animation, particle systems, post-processing.
Modes of Study	Practical assignments 36 h. Self-study 36 h. Total 72 h.
Evaluation	Passed|Failed. Credit test 30%, practical assignments 70%.
Study Materials	Materials are delivered/announced during classes.
Prerequisites	Computer graphics basic course.

DV.2.01.01	***GAME PLATFORM DEVELOPMENT 3 ECTS cr***
Year and Semester	Year 2 Semester 3
Teacher(s)	Pavel Kostenetskiy, Candidate of Science, Associate Professor of System Programming Department Gorskih Alexander, Master of Science, Assistant of System Programming Department
Aims	The student obtains basic skills in game development process. Upon completion of the course, the student will be able to design and develop game frameworks.
Content	Scene editor development: review, architecture, graphics system, particle system, resource storing system, draw-call, scripts system, AI system integration. Animation system: morphing, skeletal animation, frame-based animation, curve-based animation, animation compression techniques. Game development: game architecture, level editor, loading resources. Scene visualization: graphics pipeline, Ubershader, material subsystem. Features of design real time visualization systems.

Modes of Study	Practical assignments 54 h. Self-study 54 h. Total 108 h.
Evaluation	2-5. Practical assignments 50 %, exam 50 %.
Study Materials	Materials are delivered/announced during classes.
Prerequisites	Influences Programming Foundations course (bachelor).

DV.2.02.01	***SOCIAL NETWORK GAME DEVELOPMENT 2 ECTS cr***
Year and Semester	Year 1 Semester 2
Teacher(s)	Mikhail Mezhenin, Master of Science, Assistant of System Programming Department
Aims	Students obtain basic knowledge of social gaming mechanics and skills of working with technologies and APIs used to develop games for social networks. Upon completion of the course, each student should present a fully functional game prototype released on one of the social networks.
Content	Introduction: social gaming phenomena, criticisms, typical genres. Game development technologies: HTML5 and Canvas element, JavaScript, IFrame-based applications, MongoDB. Social gaming platforms: VK API, Facebook Graph API. Monetizing and support: balance calculation, in-game purchases, advertising, metrics.
Modes of Study	Practical assignments 36 h. Self-study 36 h. Total 72 h.
Evaluation	Passed|Failed. Credit test – 30%, practical assignments – 70%.
Study Materials	Materials are delivered/announced during classes.