**DEPARTMENT CATALOG**

**of selective academic disciplines**

**of the first (bachelor) level of higher education for educational programs of specialties:**

**121 Software engineering**

**123 Computer engineering**

# Instruction for catalog users

1. The number of selective subjects that a student can choose is determined by the curriculum. The curriculum also indicates the semester in which the selective discipline is taught, the form of semester control - assessment, types and volumes of educational classes, the amount (in ECTS credits) - 4 credits.
2. The direct choice of disciplines by students is carried out through willpower. Each student fills out a form in the "Electronic Campus" system, in which he indicates the disciplines he wishes to study in the next academic year (taking into account the number of disciplines and the study semester specified in the curriculum).
3. The catalog of selective disciplines is common to students of the 2nd and 3rd year of specialties 123 and 121 of the Department of Computer Engineering (FICE), so you need to use the information from the "Requirements before starting studies" section of the annotations and choose disciplines according to your level of training.
4. In case of impossibility of formation of educational groups of standard size for study of a specific selective discipline, students are given the opportunity to re-select within a specified period by joining the already formed educational groups.
5. At the request of a student who has chosen a certain elective discipline, he/she is allowed to join a group in which this Discipline is taught within the framework of another educational program.
6. A student cannot choose the same academic discipline twice.
7. If, for a valid reason, the student was unable to choose the disciplines on time, or discovered an error regarding his will, he applies to the dean's office with an application to enroll in the studies of his chosen disciplines, submitting the relevant documents. A student who has neglected his right to choose will be enrolled to study those disciplines that the head of the graduating department deems necessary to optimize study groups and streams.
8. The academic disciplines chosen by the student are indicated in his individual study plan.
9. More information on the procedure for students to exercise their right to freely choose academic disciplines can be found in the relevant Regulation on the procedure for students of higher education to exercise their right to freely choose academic disciplines.

# 

**CONTENT**

[Instruction for catalog users 124](#_Toc127050490)

[Annotations of selective disciplines 126](#_Toc127050491)

[Disciplines taught by the department of CE 126](#_Toc127050492)

[**AGILE software development methodology** 126](#_Toc127050493)

[**Introduction to Data Science technology** 127](#_Toc127050494)

[**Client software development technologies (Front-end)** 129](#_Toc127050495)

[**Server software development technologies (Back-end)** 130](#_Toc127050496)

[**C/Embedded programming technologies** 132](#_Toc127050497)

[**Introduction to artificial intelligence** 133](#_Toc127050498)

[**Fundamentals of software development on the Node.JS platform** 134](#_Toc127050499)

[**Fundamentals of software development on the Java platform** 134](#_Toc127050500)

[**Development of system programs** 135](#_Toc127050501)

[**FPGA Programming Technologies** 136](#_Toc127050502)

[**Program and data security systems** 138](#_Toc127050503)

[**Computer Vision Technologies** 139](#_Toc127050504)

[**Python programming technologies** 141](#_Toc127050505)

[**Distributed information systems** 142](#_Toc127050506)

[**Parallel and distributed computing** 143](#_Toc127050507)

[**Hybrid computer systems** 144](#_Toc127050508)

[**Management of IT infrastructure projects** 145](#_Toc127050509)

[**Digital signal processing** 147](#_Toc127050510)

[**Mathematical foundations of data protection and information security** 147](#_Toc127050511)

[**Mobile computer networks** 149](#_Toc127050512)

[**Basics of computer modeling** 150](#_Toc127050513)

[**System programming in the Unix environment** 151](#_Toc127050514)

[**Systems of automated design of computer systems** 152](#_Toc127050515)

[**Testing and quality control (QA) of embedded systems** 153](#_Toc127050516)

[**Methodology and organization of scientific research** 155](#_Toc127050517)

[**Artificial intelligence methods and technologies** 156](#_Toc127050518)

[**Basics of data science** 157](#_Toc127050519)

[**Statistical methods of machine learning** 158](#_Toc127050520)

[**Java programming language** 159](#_Toc127050521)

[**Data analysis using the Python language** 161](#_Toc127050522)

[**Methods and technologies of Artificial Intelligence** 162](#_Toc127050523)

[**Software tools for designing and implementing neural network systems** 162](#_Toc127050524)

# Annotations of selective disciplines

## **Disciplines taught by the department of CE**

|  |  |
| --- | --- |
| **Discipline** | **AGILE software development methodology** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Software engineering, Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Architecture  computers |
| **What will be studied?** | * the basics of terminology and basic concepts. Basic software development methodologies. Methods of selecting technologies for development. Overview of existing technological solutions used to create projects * the basics of applying theoretical knowledge on the example of developing documentation of an educational project personalized for each student or group of students of their choice |
| **Why is it interesting/necessary to study?** | * Existing needs and trends in the labor market show that for successful work as a programmer, one needs not only to be able to write code (program), but also to understand the principles of software development in general. Up to 50% of the time of a technical interview is spent precisely on issues related to development methodologies * since the dominant software development methodology is AGILE at the moment, gaining an understanding and practice of developing your own project from 0 will allow you to sufficiently prepare for future work |
| **Why you can learn (results**  **teaching)** | Understand the real process of software development. Prepare for what will be expected of the student after employment in a real workplace. |
| **How can you use acquired knowledge and skills?** | Understand how to read and write project documentation for the project as a whole or its individual components. Choose methodologies and technologies for the implementation of the project or its component. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, laboratory work |
| **Terminal control** | Offset |
| **Teacher** | Sr. teacher Shevelo Oleksiy Pavlovich |

|  |  |
| --- | --- |
| **Discipline** | **Introduction to Data Science technology** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Data structures and algorithms, Algorithms and calculation methods, System programming, System software, Organization of databases, Introduction to the Linux OS, Programming, Computer architecture, Computer networks, Discrete mathematics, Computer logic, Software engineering |
| **What will be studied?** | The purpose of the discipline is to provide a comprehensive, thorough theoretical basis and powerful practical skills in software implementation of methods, mathematical models and algorithms of Data Science technological processes (data research): data - information - knowledge - knowledge manipulation - visualization.  The specificity of the course consists in considering, along with classic Data Science methodologies, advanced author's developments obtained during the implementation of practical R&D projects.  The theoretical foundations of Data Science are provided in the form of lectures with a mandatory demonstration of the considered algorithms in the form of examples of program code.  Practical skills in the application of Data Science technologies are acquired in laboratory classes. At the same time, special attention is paid to software engineering processes.  The practical part of the discipline is focused on the application of the high-level Python programming language with the study of the functionality of the libraries: Pandas, NumPy, Matplotlib, scikit-learn.  The discipline reveals the essence of Data Science technological processes: data processing for the purpose of obtaining information - information processing for the purpose of discovering knowledge - using skills in practice - visualization of results.  The discipline consists of two interrelated units:   1. Methodological foundations of Data Science:    1. Applied statistical data analysis (data models; statistical analysis of experimental sample characteristics; processing of anomalous measurements; evaluation, extrapolation and interpolation by trend models - recurrent smoothing and smoothing of the accumulated sample; construction of nonlinear models of experimental data using differential transformations);    2. Multi-criteria decision-making methods (multi-criteria evaluation; multi-criteria identification; multi-criteria distribution of resources; multi-criteria structural and parametric synthesis of systems);    3. Intelligent data analysis. (Technologies: OLAP, Data Mining, Text Mining, Image Mining, Knowledge discovery, Speech and language recognition);    4. Application of artificial intelligence for Data Science technologies (artificial neural networks; multi-criteria optimization neural networks; model methods and algorithms of self-organization and situational analysis). 2. Technological aspects of Data Science:    1. Algorithms and technologies for prediction the dynamics of changes in financial and stock markets (according to statistical and alternative models);    2. Algorithms and technologies for determining credit risks for banking CRM systems (scoring and multi-criteria models);    3. Algorithms for identifying current situations for production CRM systems and critical infrastructure facilities (multifactor analysis and Computer Vision technologies). |
| **Why is it interesting/necessary to study?** | Mastering the knowledge, skills and abilities required for the positions: Data Scientist; Data Engineer; Data Analyst; Risk Team, etc. |
| **What will be learned?** | 1. Knowledge and skills of application and implementation of basic Data Science algorithms:  * applied statistical data analysis; * multi-criteria decision-making methods; * intelligent data analysis; * application of artificial intelligence for Data Science technologies; * visualization of Data Science results.  1. Knowledge and skills of using basic Python libraries to implement Data Science algorithms: Pandas, NumPy, Matplotlib, scikitlearn. |
| **How can you use acquired knowledge and skills?** | Acquired competencies in Data Science can be applied to projects in the following applied industries and technologies:   1. Data analysis for e-commerce tasks; 2. Data analysis for industrial and infrastructure CRM systems; 3. Analysis of visual and geospatial data of various directions; 4. Data analysis for medical information systems. |
| **Information support** | Educational and working programs of the discipline, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich |

|  |  |
| --- | --- |
| **Discipline** | **Client software development technologies (Front-end)** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Algorithms and calculation methods, System programming, System software, Algorithms and Data structures, Organization of databases, Introduction to the Linux OS, Programming, Computer architecture, Computer networks, Discrete mathematics, Computer logic, Software engineering, Computer systems |
| **What will be studied?** | The purpose of the discipline is to provide thorough theoretical knowledge and specific practical skills in the development of Front-end (client) software.  The educational discipline is aimed at the formation of competencies in the development of the Front-end part of applied Web-applications. In combination with the educational discipline «Technologies for the development of server software (Back-end)» (7th semester), the acquisition of comprehensive knowledge and practical skills in the design and development of application software, based on modern and promising client-server technologies of distributed processing and information exchange, is ensured.  The theoretical and practical components of the educational discipline are aimed at acquiring competencies during the implementation of a specific project on the development of the Front-end part with simplified implementation of the Back-end component. |
| **Why is it interesting/necessary to study?** | Mastering the knowledge, skills and abilities required for the positions: Front-end developer, Front-end Engineer, Front-end Team Lead, DevOps engineer, etc. |
| **What will be learned?** | 1. Knowledge and skills in the application and implementation of client service technologies to create effective front-end parts of modern Web applications:  * front-end interface prototyping; * user interface programming technologies.  1. Knowledge and skills of using basic Front end technologies: HTML5; CSS3; JavaScript; TypeScript; jQuery; React and Flux architecture; Angular 6 and MVC architecture. |
| **How can you use acquired knowledge and skills?** | Acquired competences in the development of Front-end software can be applied to projects of the following applied industries and technologies:   1. E-commerce; 2. Electronic document management; 3. CRM systems of various applied directions, built on technologies of distributed processes of collection, storage, processing and exchange of information. |
| **Information support** | Educational and working programs of the discipline, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich |

|  |  |
| --- | --- |
| **Discipline** | **Server software development technologies (Back-end)** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Data structures and algorithms, Discrete mathematics, Software engineering. |
| **What will be studied?** | The purpose of the discipline is to provide thorough theoretical knowledge and specific practical skills in server software development.  The educational discipline is aimed at the formation of competencies in the development of the Back-end part of applied Web-applications. In combination with the educational discipline «Technologies of programming user interfaces (Front End)» (8th semester), the acquisition of comprehensive knowledge and practical skills in the design and development of application software based on modern and promising client-server technologies of distributed processing and information exchange is ensured.  The theoretical and practical components of the educational discipline are aimed at acquiring competencies during the implementation of a specific project on the development of the Back-end part with a simplified implementation of the Front-end component.  The discipline consists of the following blocks:   1. Methodology and technologies of building and creating client-server (Web) applications (model «client-server», cloud services, network level, exchange protocol and Internet standards: TCP/IP, WWW, XML, JSON, HTTP, HTTPS, FTP, Telnet, MIME; URL and URI; the main tools of errors research (ICMP, ping, traceroute); sockets, IP and port’s addressing, using proxy-servers; name search services: DNS, whois; remote access services: Telnet, SSH, Remote Desktop, VNC); 2. API and its protocols. Organization of the access restriction and authorization system (rules). Frameworks and toolset for Back-end development. Interaction with the database by using the object manager (ORM – Hibernate, Doctrine, typeORM, Eloquent). Technologies of back-end software development: PHP (Symphony, Laravel), Java (Spring Boot, Java EE), JavaScript (Express, NestJS); 3. Specification of web-services. Code testing (OPENAPI3 documentation; creating of schemes; basics of Test Driven Development – unit, integration and e2e tests (PHPUnit, JUnit, Jest)); catching and handling errors. 4. Introduction to DevOps. Component parts of DevOps: Development (Software Engineering), Operations, Quality; Assurance; configuration of services Apache, PHP, MySQL; DevOps integration into the development process of software; DevOps and Digitalization; CI/CD development; High Availability. |
| **Why is it interesting/necessary to study?** | Mastering the knowledge, skills and abilities required for the positions: Back-end developer; Back-end Engineer; Back-end Team Lead; DevOps Engineer, etc. |
| **What will be learned?** | 1. Knowledge and skills of client-server application and implementation technologies for creating effective Back-end parts of modern Web applications:  * methodology and technologies of construction and creation of Back-end parts client-server applications; * modern server software development technologies; * code testing; * basics of implementing DevOps tasks.  1. Knowledge and skills of using basic Back-end approaches: frameworks, testing, API development. |
| **How can you use acquired knowledge and skills?** | Acquired competences in the development of server software (Back-end) can be applied to projects of the following applied industries and technologies:   1. E-commerce; 2. Electronic document management; 3. CRM systems of various applied directions, built on technologies of distributed processes of collection, storage, processing and exchange of information. |
| **Information support** | Educational and working programs of the discipline, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich |

|  |  |
| --- | --- |
| **Discipline** | **C/Embedded programming technologies** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Dicscrete mathematics, Computer logic, Computer arithmetic, Programming, Computer architecture, Introduction to the Linux OS. |
| **What will be studied?** | Creating software for ARM-based microcontrollers. Implementation of software configuration management and development of embedded programs. Deploying a processor core using Toolchain. Using the Git version control system. Software development in Linux on a virtual machine. Development of a project with its own firmware based on hardware for programming and debugging microcontrollers. Using BeagleBone Black, Raspberry Pi, STM StarterKit GlobalLogic platforms. |
| **Why is it interesting/necessary to study?** | Embedded systems and IoT devices require software to function and achieve their operational purpose. However, writing this software from scratch for each type of hardware is not practical. The most common approach is the use of operating systems to solve the main task of managing hardware resources and providing services for user applications. Field of application. Development of solutions for the Internet of Things (IoT), embedded systems, smart systems. Development of hardware systems for the automotive industry, industrial automation, high-performance computing, artificial intelligence and machine learning. |
| **What will be learned?** | * Install and configure the Linux operating system (Linux kernel, U-Boot, BusyBox) for the architecture of ARM processors by compiling the source code. * Program in the Bash shell. * Use the Git version control system * To program in C for Linux OS for processors of the ARM architecture * Program your own modules for the processor core * Perform testing of own product on the basis of modern platforms for development. * Work in a team. |
| **How can you use acquired knowledge and skills?** | The knowledge gained is sufficient to design the architecture and implement your own smart device, debug it and configure it to work in an IoT environment.  The experience and practice gained are sufficient to continue independent professional training in the field of developing new IoT systems using any hardware and software available on the market.  The acquired experience, knowledge and practice are sufficient to successfully pass interviews in IT companies engaged in the development of embedded systems and IoT. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, practical and seminar classes, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Assistant Kaplunov Artem Volodymyrovych |

|  |  |
| --- | --- |
| **Discipline** | **Introduction to artificial intelligence** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian, English |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Algorithms and calculation methods, System programming, System software, Algorithms and Data structures, Organization of databases, Programming, Computer architecture, Computer networks, Discrete mathematics, Software engineering, Parallel programming |
| **What will be studied?** | Basic knowledge about the context, terms, problems and current state of research in the field of artificial intelligence. Overview of the main principles and approaches to the study of artificial intelligence systems. An overview of the main areas of research and relevant artificial intelligence tools: genetic algorithms, evolutionary methods, machine learning, deep learning, reinforcement learning, etc. Overview of the main models and algorithms with examples of their application. |
| **Why is it interesting/necessary to study?** | In recent years, the field of artificial intelligence has experienced a period of real uplift due to the rapid development of new models of neural networks, software tools and the rapid development of computing resources based on graphics (GPU) and tensor (TPU) accelerators. Research on artificial intelligence methods is currently showing the fastest growth rates in scientific research and wide practical application in many areas of life around us: from personal assistants in smartphones to self-driving cars. |
| **What will be learned?** | Theoretical knowledge and basic practical experience in the application of various methods of artificial intelligence to existing practical problems in a wide range of applications. |
| **How can you use acquired knowledge and skills?** | The acquired knowledge will allow to understand the place and role of artificial intelligence methods in the general context of information technologies. This discipline is a necessary stage for preparing for the professional level of mastering specialized knowledge and skills in the field of artificial intelligence, which will be taught in the following educational disciplines, which are devoted to a more detailed study of individual methods of artificial intelligence. |
| **Information support** | Educational and working programs of the discipline, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Hordienko Yury Hryhorovych |

|  |  |
| --- | --- |
| **Discipline** | **Fundamentals of software development on the Node.JS platform** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-Oriented Programming, Algorithms and Data structures, Computer networks. |
| **What will be studied?** | Development of programs in the JavaScript language. Studying of basic principles, basic constructions of language. Using the Node.Js platform to develop web systems. Development tools for server components and front-end applications. |
| **Why is it interesting/necessary to study?** | Mastering modern technologies for the development of web systems for the creation of multi-platform software. |
| **What will be learned?** | Create effective web systems software. |
| **How can you use acquired knowledge and skills?** | Ability to create software for a variety of OSs and hardware platforms. Ability to create application software for computer systems and networks. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Sr. teacher Shemsedinov Timur Gafarovych |

|  |  |
| --- | --- |
| **Discipline** | **Fundamentals of software development on the Java platform** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-Oriented Programming, Algorithms and Data structures, Computer Discrete mathematics. |
| **What will be studied?** | Principles of construction of the Java language, basic structures, libraries. Implementation of the principles of object-oriented programming, modularity, working in the network and with databases. Classes, interfaces, packages and other basic elements of building programs. Program development tools, integrated software development and debugging environments. Java Virtual Machine. Organization of multithreaded execution. |
| **Why is it interesting/necessary to study?** | Mastering modern Java technologies for creating platform-independent software, technologies for developing complex information systems for solving a wide range of tasks. |
| **What will be learned?** | Create effective software based on Java technologies, learn and master modern software development tools based on an object-oriented approach and popular design and programming patterns. |
| **How can you use acquired knowledge and skills?** | Ability to build software for a variety of operating systems and hardware (network, mobile, embedded) platforms. Ability to create application software for computer systems and networks. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Sr. teacher Aleshchenko Oleksiy Vadimovych |

|  |  |
| --- | --- |
| **Discipline** | **Development of system programs** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Introduction to the Linux OS |
| **What will be studied?** | * – composition of system software of modern computer systems; * – technology of step-by-step development of complex software products * Incremental Approach on the example of developing system programs; * – practical application of this technology during the development of system programs; * – offseting of developed system programs. |
| **Why is it interesting/necessary to study?** | * system software is present in any computing device, so the demand for those who understand it * will always be high; * system software is the basis of the functioning of computer systems, it is it that dictates the requirements for application programs, therefore its developers will always be ahead of programmers applicants; * the development of system software is usually carried out by the most well-known companies that determine the trend of computer technology development for many years to come. |
| **What will be learned?** | * develop and Offset system programs, understand messages   operating systems, perform code optimization. |
| **How to use acquired knowledge and skills**  **(competence)** | * in the development and improvement of operating systems, specialized software systems, computer security systems, programming systems, individual components of system software: drivers, file management programs, etc. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, laboratory work, independent work. |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Pavlov Valery Georgiyovych |

|  |  |
| --- | --- |
| **Discipline** | **FPGA Programming Technologies** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of computer logic, computer arithmetic, discrete mathematics. Basic knowledge of computer architecture, including knowledge of the general structural and functional organization of Von-Neumann architecture computers, general principles of firmware management, processor core structure, and general principles of firmware level organization. Skills  programming. |
| **What will be studied?** | Digital design and engineering methods for FPGAs. Hardware description languages for the development of digital devices. Design of digital devices using Verilog. Functional modeling and structural synthesis in modern CAD.  Practical tasks are initially based on simple examples of programming computer hardware components. Next, an introduction to the main stages of the full cycle of development of IP-cores of digital technology is offered, including functional synthesis, structural synthesis, verification of developments and programming of the FPGA microcircuit. The basic issues of creating OffsetBench levels in the Verilog language are considered. CAD models ModelSym MentorGraphic, Quartus II are used to check the correctness of the created projects and debugging. Issues of integration of CAD Quartus II and ModelSim are considered. Altera DE2 Board (Cyclon II), DE10 Standard Board is used for experiments  (Cyclon IV), DE1 SoC Board (Cyclon V). |
| **Why is it interesting/necessary to study?** | Programmable logic (FPGA) has recently become the main technology used to create electronic systems in various fields of application. The acquired skills and knowledge will be relevant in the areas of developing solutions for the Internet of Things (IoT), embedded systems, smart systems. Development of hardware systems for the automotive industry, industrial automation, high-performance computing, machine artificial intelligence  teaching. |
| **What will be learned?** | Design digital circuits and devices using modern design methods and hardware design languages. Gain programming skills in Verilog. Use modern professional tools for digital design and modeling. Perform functional modeling and debugging of devices in ModelSym CAD. Perform structural synthesis in CAD Quartus II Altera. Implement devices on modern Altera DE2 boards  Board, DE10 Board, DE1 SoC Board. |
| **How to use acquired knowledge and skills**  **(competence)** | Basic knowledge in the use of the hardware design language, Verilog, as well as experience and skills in the use of modern CAD and Altera products is sufficient for continuing professional training in the direction of hardware design for complex systems on FPGA and ASIC, system-on-chip (SoC), embedded systems, smart systems. software and hardware implementations of artificial intelligence systems.  The acquired basic knowledge of programming and digital design methods will allow, if necessary, to quickly switch to the use of hardware programming language VHDL, chips of other FPGA manufacturers (Xilinx), as well as other CAD for functional and structural synthesis on FPGA and ASIC (Synopsys, Aldec, Cadence).  The acquired basic knowledge will help in further professional work  learning, including independently. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, practical and seminar classes, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, associate professor, professor Klymenko Iryna Anatoliivna |

|  |  |
| --- | --- |
| **Discipline** | **Program and data security systems** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Computer architecture, Computer networks, Algorithms and Data structures, Programming, Discrete mathematics, Probability theory and Mathematical statistics. |
| **What will be studied?** | This course is devoted to an in-depth study of specific issues related to the mathematical and algorithmic aspects of information protection in computer systems and the design and development of secure software and security systems. Methods and algorithms of protection during data transmission and storage, development of software and hardware means of protection are studied. In particular, protected network protocols, encryption methods, cryptography, and steganography. The issue of creating and organizing secure services using cloud technologies in various industries, in particular, e-commerce, e-banking, functioning of cryptocurrencies. |
| **Why is it interesting/necessary to study?** | The modern realities of functioning of the digital society require using and develop computer tools and systems resistant to unauthorized access and use of programs and data. Every qualified specialist related to software must know the principles of information security and methods of protecting programs and data and be able to apply them competently. |
| **What will be learned?** | Learn to make decisions about the choice of data protection structure; understand mathematical methods and algorithms of cryptography and data encryption.  Develop information protection software that is resistant to hacker attacks. Organize secure storage, processing and transmission of information using technologies of distributed and cloud systems. |
| **How can you use acquired knowledge and skills?** | The ability to analyze, select, develop and apply methods and tools to ensure information security, cybersecurity. The ability to perform work as an analyst of computer systems and software. Being a qualified software developer and user of hardware and software complexes. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Volokita Artem Mykolayovych |

|  |  |
| --- | --- |
| **Discipline** | **Computer Vision Technologies** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Algorithms and calculation methods, System programming, System software, Algorithms and Data structures, Organization of databases, Programming, Computer architecture, Computer networks, Discrete mathematics, Computer logic, Software engineering, Parallel programming |
| **What will be studied?** | The purpose of the discipline is to provide a thorough theoretical basis and powerful practical skills in software implementation of methods, mathematical models and algorithms of Computer Vision technologies.  The theoretical foundations of Computer Vision are provided in the form of lectures with a mandatory demonstration of the practical implementation of the considered algorithms in the form of program code examples.  Practical skills in the application of Computer Vision technologies are acquired in laboratory classes, which are built on the principle of increasing the functionality of developed scripts. At the same time, special attention is paid to software engineering processes.  The practical part of the discipline is focused on the application of the high-level Python programming language with the study of the capabilities of the graphics libraries Graphics, Tkinter, Matplotlib, NumPy (for the "raw" implementation of Computer Vision algorithms) and specialized packages such as PIL/Pillow, OpenGL, OpenCV for creating software modules of a completed practical orientation.  The discipline reveals the essence of the stages of the classic pipeline of processing digital graphic images: image synthesis (spatial transformation) - rasterization (realistic image, digital processing) - vectorization (object identification and target image processing). This is implemented in the following topics:   1. Digital image synthesis (2D, 3D - objects, graphic method, analytical and non-analytical methods); 2. Basic geometric transformations of 2D, 3D objects (scaling, moving, rotating, projections); 3. Rasterization and raster images, creation principles, characteristics and processing algorithms (rasterization, brightness change, filtering); 4. Vectorization and vector images, mathematical models, approximation, interpolation and smoothing algorithms (MNC, Splines, Bezier curves, etc.); 5. Realistic image models, augmented reality, light and shadow models: 3D pipeline, rendering; 6. Digital image processing: filtering; color correction; segmentation; vectorization; morphological processing; recognition; 7. Methodological, technological and applied aspects of Computer Vision: sensor data processing; navigation; Artificial Intelligence; management. |
| **Why is it interesting/necessary to study?** | Mastering the knowledge, skills and abilities required for the positions: Software Developer with Computer Vision; Embedded Developer for Computer Vision systems; Computer Vision Research Engineer, etc. |
| **What will be learned?** | 1. Knowledge and skills of application and implementation of basic Computer Vision algorithms:  * synthesis of 2D, 3D graphic objects and their geometric transformations; * synthesis and processing of raster images; * synthesis and processing of vector images; * formation of realistic images; * digital image processing; * applied aspects of Computer Vision; * comprehensive implementation of stages of the digital image processing pipeline for Computer Vision tasks and other applied fields.  1. Knowledge and skills of using basic Python libraries to implement Computer Vision algorithms: Matplotlib, NumPy PIL/Pillow, OpenCV. |
| **How can you use acquired knowledge and skills?** | Acquired competencies in Data Science can be applied to projects in the following applied industries and technologies:   1. 3D scene reconstruction and navigation in it (Structure-from-Motion, Road Scene Understanding and Autonomous Driving); 2. Scene segmentation and plot understanding (Significance maps, video and 3D segmentation, 3D stream, multiple tracking, object detection, activity detection and prediction, group analysis, object detection and recognition); 3. Image recognition, object identification (Convolution networks, Deep learning, Image Retrieval, Object Detection). |
| **Information support** | Educational and working programs of the discipline, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich |

|  |  |
| --- | --- |
| **Discipline** | **Python programming technologies** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Software engineering, Basics of parallel programming |
| **What will be studied?** | Methods of converting project documentation into a real code. Methods of process optimization for writing code to achieve tasks according to trade-off strategy, working with risks, etc.  Skills of allocating tasks in time and planning a full development cycle.  Code testing.  Code deployments on server infrastructure. |
| **Why is it interesting/necessary to study?** | One of the major issues of modern software is the need not only to write high-quality code, but also the ability to meet deadlines without significant quality losses. It is also necessary to be able to work with risks, priorities, etc. |
| **What will be learned?** | The acquired knowledge will allow you to learn how to transform a project documentation into a real code, meeting the requirements both in terms of time and code quality. Working in the development team. |
| **How can you use acquired knowledge and skills?** | To develop software in a team or independently. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, practical and seminar classes, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Sr. teacher Shevelo Oleksiy Pavlovich |

|  |  |
| --- | --- |
| **Discipline** | **Distributed information systems** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Software engineering, Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Introduction to the operating room  Linux systems, Computer architecture, Discrete mathematics, |
| **What will be studied?** | * architectural templates for the implementation of distributed information systems, the basics of service-oriented architecture, principles and protocols of interaction of web services, principles of building software interfaces, ways of integrating distributed software resources; * basics of software interaction with SQL and NoSQL databases, concepts of streaming data processing; * the basics of technological equipment for the process of designing, documenting, developing and Offseting software; * basics of building web applications (Single Page Application, SPA, Power Web Application, PWA) using modern software tools (React, Angular, Vue); * ways of implementing the process of collecting and placing software * systems using PAAS cloud services. |
| **Why is it interesting/necessary to study?** | Today, the possession of knowledge, abilities and skills related to the process of designing distributed systems of collective use, the process of developing appropriate software and organizing technological support for all stages of the software life cycle is one of the key  competitive advantages in the IT labor market. |
| **What will be learned?** | Carry out the design of information systems based on the use of modern architectural principles with the use of modeling and documentation tools at all stages of the software life cycle, develop information and software for such systems and organize technological support the process of their development and placement. |
| **How to use acquired knowledge and skills (competencies)?** | Carry out design and development of modern information systems for collective use, organize technological support for the software development process, use tools for organizing distributed data processing, use relational, document-oriented and graph databases, integrate own software with publicly available web services, place software in  PAAS cloud environment. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, laboratory work |
| **Terminal control** | Offset |
| **Teacher** | Ph.D., associate professor Boldak Andriy Oleksandrovich |

|  |  |
| --- | --- |
| **Discipline** | **Parallel and distributed computing** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Computer architecture, Aganalytic geometry, Engineering  Software, |
| **What will be studied?** | Methods of organizing calculations in parallel computer systems with different structural organization: systems with shared memory, systems with local memory, distributed systems. The life cycle of program development for a parallel system, which includes:   * development of a parallel calculation algorithm. * development of flow execution algorithms with a solution to the task of mutual exclusion and synchronization. * development of a structural diagram of flow interaction for selected parallel programming languages (libraries) and means of flow organization: semaphores, mutexes, events, critical sections, * methods of debugging and Offseting a parallel program * research of the percentage of loading of processors (cores) of the system, determination of execution time, calculation of the acceleration factor (speedup).   Modern parallel programming languages and libraries are used in laboratory works: Java, C#, WinAPI, Ada, OpenMP, MPI,  TBB. |
| **Why is it interesting/necessary to study?** | Modern computers (from tablets to supercomputers) are equipped with multi-core processors, so creating efficient software for them requires the use of threads for optimal use of multi-core hardware. Multi-core processors are also part of most gadgets, such as smartphones, mp3 players, video players, game consoles, which allows you to speed up (improve) sound processing and image. |
| **What will be learned?** | Ability to develop software for parallel and distributed computer systems of various purposes, analyze tasks for the presence of parallelism, build optimal parallel algorithms, use threads to build a program, organize optimal interaction of threads depending on the structure of the computer system, debug a parallel program, prevent and get rid of deadlocks.  Use of modern parallel programming languages and libraries. |
| **How to use acquired knowledge and skills (competencies)?** | The acquired basic knowledge will help in further professional training, including independently, expand the perspective of career opportunities in the field of software development for any parallel computer systems, real-time systems, and as well as creation of server-client applications. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, practical and seminar classes, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Korochkin Oleksandr Volodymyrovych |

|  |  |
| --- | --- |
| **Discipline** | **Hybrid computer systems** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Probability theory and mathematical statistics, System programming. System software, Computer systems, Computer architecture |
| **What will be studied?** | Fundamentals of designing analog and hybrid operating units, analog-to-digital and digital-to-analog information converters, analog and hybrid processors, methods of mathematical modeling using operating units, methods of preparing problems for analog and hybrid systems, programming methods for analog and hybrid computer systems |
| **Why is it interesting/necessary to study?** | The acquired knowledge allows mastering the basic methods of building specialized computer systems |
| **What will be learned?** | Using of modern technical and software design tools, analog and hybrid computer systems |
| **How can you use acquired knowledge and skills?** | Hybrid CSs will allow to effectively solve the following tasks:   * simulation of automatic control systems in real time that contain analog and digital devices; * real-time reproduction of processes containing high-frequency components and variables in a wide amplitude and frequency range; * statistical modeling; * modeling of biological systems; * solutions of equations in partial derivatives; * optimization of management systems. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Selivanov Viktor Levovych |

|  |  |
| --- | --- |
| **Discipline** | **Management of IT infrastructure projects** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Computer architecture, Discrete mathematics, Computer logic, Software engineering, Basics of parallel programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Computer networks |
| **What will be studied?** | The main goal of the discipline is the acquisition of theoretical knowledge and practical skills of project management and teamwork: gaining of soft skills by applying one's hard skills to solving practice problems. It is implemented in the form of organization and implementation of a specific startup project in the field of information technology and computer engineering. The content of the discipline takes into account the best practices of leading IT enterprises and involves the involvement of mentors - practitioners. The result of working out the tasks of the discipline is a completed practical project.  The discipline involves the study of two interrelated components.   1. Project management:   Organization and implementation of the pre-project R&D stage; development technical proposal and technical task; requirements engineering (business analysis); architectural design; system development (software); quality assessment (verification and testing); implementation and basics of DevOps; accompaniment; organization of processes (life cycle models) according to classic and Agile models; reporting and submission of the project.   1. Group dynamics:   Theory and practice: professions and positions in IT enterprises and their scope of responsibilities; drafting an effective resume; conducting an interview; selection of talented personnel; formation of groups and teams, organization of teamwork; leadership; conflictology; there-management; preparation, organization and conducting of effective meetings, presentations, negotiations. |
| **Why is it interesting/necessary to study?** | The acquired knowledge and skills ensure understanding, participation and implementation of modern processes of team development of complex hardware and software systems and complexes at the level of advanced national and world practices. |
| **What will be learned?** | Knowledge and skills of organization, application and implementation:   * team executing of projects using all phases of the life cycle (from gathering requirements to implementation); * execution of projects using different development methodologies (classical and Agile) and in different roles (developer, tester, analyst, DevOps, manager, etc.); * using of automated tools for team development of projects; * business communications (meetings, presentations, active listening, written communication, interviewing, etc.); * teamwork (giving feedback, sharing experiences, resolving conflict situations, etc.); * time management; * practical solutions to problem situations; * search, collection and processing of information necessary for the implementation of practical tasks. |
| **How can you use acquired knowledge and skills?** | Team work on complex technical projects in accordance with modern requirements for processes and technologies for their implementation. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich |

|  |  |
| --- | --- |
| **Discipline** | **Digital signal processing** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Higher mathematics, Programming, Discrete mathematics, Computer architecture, Computer systems |
| **What will be studied?** | Mathematical foundations and algorithms of digital signal processing (DSC) and images, as well as methods of their assembly, programming and hardware implementation. |
| **Why is it interesting/necessary to study?** | Modern hardware and software tools that are common in the IT industry are very often based on the implementation of COS algorithms. These are, for example, means of automation, telecommunications, image processing, computer games, pattern recognition and artificial intelligence, IoT. Without knowledge of TOS algorithms, it is almost impossible to Offset and modernize and develop such tools. |
| **What will be learned?** | * to choose, model and modernize the algorithms of the TSO; * to develop applications that execute TSO algorithms. |
| **How to use acquired knowledge and skills (competencies)?** | * to carry out analysis, diagnostics, selection of existing COS means, * carry out the modernization of existing and development of new means of COS, * to implement COS algorithms in IoT and artificial intelligence tools. Knowing the basics of COS and having experience in its programming is a mandatory requirement for employment in most companies involved in the development and distribution of applications and automation equipment. * telecommunications, artificial intelligence, IoT. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, laboratory work |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Sergienko Anatoly Mykhailovych |

|  |  |
| --- | --- |
| **Discipline** | **Mathematical foundations of data protection and information security** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Computer architecture, Computer systems and networks, Algorithms and Data structures, Discrete mathematics, Probability theory and Mathematical statistics, System programming |
| **What will be studied?** | Mathematical foundations of modern cryptography: irreversible transformations of Boolean algebra, number theory, elliptic curves and finite Galois fields. Methods of synthesis of cryptographic Boolean transformations, construction of data protection algorithms based on them. Modern algorithms and protocols of information protection based on Boolean functions. Attacks on algorithms and protocols of this class. Methods of building cryptographic protection systems based on irreversible transformations of number theory. Algorithms and protocols of information protection based on irreversible transformations of number theory and methods of breaking them. Mathematical foundations of the organization of homomorphic encryption of data during their processing in the clouds. Methods of breaking data protection tools by means of temporal or static analysis of power consumption dynamics. Mathematical principles of software polymorphism in the protected organization of calculations. Quantum computing and quantum cryptography. |
| **Why is it interesting/necessary to study?** | The progressive development of information integration will require qualitatively new means of data protection and demarcation of access to them in integrated environments. Creating and using such tools is impossible without mastering the fundamentals of cryptography. Professional analysis of the real level of security of data and programs requires special mathematical training. |
| **What will be learned?** | To gain thorough knowledge in special sections of mathematics, which are the basis of modern and promising means of cryptographic data protection. Master the skills of creating basic blocks of cryptographic data protection, building algorithms and protocols based on them. Acquire the ability to analyze the level of protection of information security tools. Know the protocols of cryptographic protection of information in computer systems and networks. |
| **How can you use acquired knowledge and skills?** | Develop, improve and adapt software and hardware tools for cryptographic data protection to the conditions of specific use. Create homomorphic ciphers to protect data directly in the process of their remote processing in the clouds, depending on its characteristics. To carry out a professional analysis of information security of data in view of the ability to resist various types of attacks. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Markovskyi Oleksandr Petrovich |

|  |  |
| --- | --- |
| **Discipline** | **Mobile computer networks** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Computer networks, System programming, Computer architecture |
| **What will be studied?** | Basic construction concepts, architecture, communication system of mobile computer networks, wireless broadband networks, wireless personal networks, wireless 5G networks, software-configured mobile networks, mobile cloud technologies, mobile Internet, basics of security of mobile computer networks. |
| **Why is it interesting/necessary to study?** | The course is aimed at the formation of theoretical knowledge and practical skills in the building, management, modernization, monitoring of modern mobile computer networks. |
| **What will be learned?** | To learn the principles of formation of theoretical knowledge and practical skills in construction, operation, monitoring and performance analysis of modern mobile computer networks. |
| **How can you use acquired knowledge and skills?** | The ability to select the type, plan and implement mobile computer networks, manage network resources, select a complex of necessary hardware and software tools for a mobile computer network, expand and modernize networks, monitor and analyze performance, diagnose and solve problems. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Kulakov Yury Oleksiyovych |

|  |  |
| --- | --- |
| **Discipline** | **Basics of computer modeling** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-oriented programming, Algorithms and computational methods, System programming, Algorithms and data structures, Introduction to the Linux operating system, Discrete mathematics, Software engineering, Theory  probabilities |
| **What will be studied?** | * Imitation, mathematical, logical, evolutionary, structural-linguistic modeling methods. Fundamentals of analog and physical modeling. Modeling methodology as a technology of scientific research and design of computer systems and their software. - Methods of creating models taking into account the requirements for their adequacy, methods and techniques of their software implementation and use in practice, methods of evaluating the results obtained with the help of models. Using models to optimize development and assess its reliability. Methods of evaluating the adequacy of models. * The most common models of computer systems in practice are Petri nets and modeling software packages. - Mathematical models of process optimization and methods of their software implementation. Dynamic and integer programming. Methods of evolutionary modeling based on genetic algorithms and technologies artificial intelligence. |
| **Why is it interesting/necessary to study?** | * acquired knowledge and practical skills make it possible to scientifically and methodically competently apply modeling to solve a wide range of practical problems of designing systems, software, organizing data transmission in local and global networks, evaluating the reliability of programs and the level of information security. * to find optimal solutions for project solutions and natural-scientific and economic problems with the help of modeling. * to analyze the behavior of systems and programs in various, including including critical situations, as well as in the presence of errors during their development. |
| **What will be learned?** | * Analyze the possibilities of solving a wide range of practical problems using modeling methods; make a justified choice of the model and its construction taking into account the requirements for its adequacy, competently assess the reliability of the obtained modeling results; * to master practical ways and techniques of programming simulation, mathematical and evolutionary models * Use mathematical models of process optimization and dynamic programming to create effective programs. * To evaluate the quality parameters of systems and programs, in particular, the reliability of their operation in various applications, using computer models * situations, as well as the level of data and program security |
| **How to use acquired knowledge and skills (competencies)?** | * To use the arsenal of computer modeling methods as a tool of scientific research in technical, natural-scientific, medical, economic and social spheres. * Apply computer modeling methods and existing software products for effective design of systems and programs taking into account the specified requirements for their quality characteristics * Optimizing project solutions using computer modeling; * Analyze the behavior of systems and programs in critical situations * situations, evaluate their reliability, search for errors. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, laboratory work |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Volokita Artem Mykolayovych |

|  |  |
| --- | --- |
| **Discipline** | **System programming in the Unix environment** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Introduction to the Linux operating system, Architecture  Computers, Software Engineering, |
| **What will be studied?** | Design and development of system programs for Unix-like systems. Programming takes place at the level of system calls (the corresponding functions in the standards of program development in the Unix environment), that is, at a low level of interaction with the kernel. Basic information on APIs in a Unix-like system (with an understanding of system call implementations in the kernel) for developing system programs. The discipline is not focused on system programming in any one specific implementation of a Unix-like operating system, that is, portable system programming will be studied. The course consists of the following topics: process environment, process management, advanced file handling, signal handling, interprocess communication (IPC), advanced I/O, terminal programming. |
| **Why is it interesting/necessary to study?** | This discipline should be studied by those who will develop system programs for Unix-like systems. Programming tasks in the discipline are performed in C or C++, but the acquired knowledge will be useful for developing system programs for Unix-like systems  in other programming languages. |
| **What will be learned?** | Develop system programs for Unix-like systems in the C or C++ programming language for managing processes, working with files, signals, using advanced I/O, interprocess  interactions (IPC), work with terminals. |
| **How to use acquired knowledge and skills (competencies)?** | The acquired knowledge can be used in the design and development of system programs for Unix-like systems, to support the source code of existing system programs for Unix-like systems, in the development of more effective applications  programs |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct**  **classes** | Lectures, laboratory work |
| **Semester control** | Offset |
| **Teacher** | Sr. teacher Simonenko Andriy Valeriyovych |

|  |  |
| --- | --- |
| **Discipline** | **Systems of automated design of computer systems** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian, English |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Higher mathematics, Programming, Discrete mathematics, Computer circuitry, Computer architecture |
| **What will be studied?** | * the basics of the structure of automated design systems (CAD) computer systems (CS); * the technology of designing specialized conveyor systems based on programmable logic integrated circuits (PLCs) using modern CAD; * design of specialized multiprocessor CPUs and their interfaces. |
| **Why is it interesting/necessary to study?** | * Modern PLD are almost the only elemental base that makes it possible to design the latest CSs, which differ in high performance, reliability, and energy consumption characteristics; * PLDs are used in medical devices, telecommunications, transport, data centers, the Internet of Things, military electronics, aerospace industry, to solve artificial intelligence problems; * there is a big shortage of professional personnel, both in Ukraine and in the world, who are able to design CS on PLD, whose salary is, as a rule, higher than the salary of an average programmer. |
| **What will be learned?** | * To compile parallel algorithms for high-performance calculations, * to program algorithms in VHDL language for implementation in hardware CSs, * to model and test CS projects, compile the CS description into PLD firmware. |
| **How can you use acquired knowledge and skills?** | * To analyze the suitability of PLD for performing computer tasks that require high performance; * to design high-performance CS on PLD according to the effective technology; * to perform modernization of existing high-performance CS on PLD; * to organize the execution of algorithms described in C, Python, on hardware accelerators on PLD.   Mastery of PLD programming technology is required in Ukraine in the fields of military equipment design, telecommunications, information protection, aircraft construction, including design of drones and abroad in these fields, in artificial intelligence systems, IoT, ADAS, robotics, design of custom microcircuits. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Sergienko Anatoly Mykhailovych |

|  |  |
| --- | --- |
| **Discipline** | **Testing and quality control (QA) of embedded systems** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Computer architecture, Software engineering, Basics of parallel programming |
| **What will be studied?** | Basics of testing. Testing purposes at different stages of life product development cycle. Software development models. Basic testing levels and their relationships. Compilation of test documentation. Working with equipment based on BeagleBone Black platform. Independent creation of and embedded system, setting up an environment. Experience with embedded operating system testing in view of test environment setup, as part of which following question are considered:   * Network troubleshooting; * Configuration of the Linux OS and computer network; * OS deployment (Linux kernel, U-Boot, BusyBox) for ARM Cortex A8 processor architecture by source code compilation; different approaches to flashing processor chips on the board; * Performing testing of own product based on BeagleBone Black platform; * Using Git and GitHub for version control;   Validation of functional and non-functional attributes software and hardware in self-created embedded system. Creating tests for verification and validation of embedded software and hardware in accordance with client requirements. |
| **Why is it interesting/necessary to study?** | In IT companies, great attention is paid to product quality, which is released to the market. The main reasons for the need for testing, which explain the special training of specialist testers:   * finding errors in software and hardware; * risk reduction for both users and the company; * reduction of development and maintenance costs; * productivity improvement.   Embedded software and hardware testing is an excellent approach to ensure safety in critical applications such as medical equipment, railways, aviation, transportation industry, etc. Large IT companies engaged in such developments invite specially trained testers to work. To train specialists for testing embedded systems, it is necessary to have basic knowledge in the sphere of design and operation of embedded systems, computer networks. |
| **What will be learned?** | * To configure the Linux OS and computer network; * To deploy an OS (Linux kernel, U-Boot, BusyBox) for ARM Cortex A8 processor architecture by source code compilation; * Use different methods of flashing processor chips on the board. Use various interfaces, terminal emulation programs for communication and debugging of built-in devices; * Perform testing of own product based on BeagleBone Black platform; * Find errors in software and hardware embedded systems; * Troubleshoot computer network problems; * Compile test documentation; * Work in team; * Get basic training for QA specialist certification ISTQB. |
| **How can you use acquired knowledge and skills?** | Work in IT as a software and hardware tester. Continue training in the sphere of software and hardware development of embedded systems. Continue training in the sphere of software and hardware testing automation specialist. Acquired basic knowledge in the field of the basics of QA testing can be used in the field of software testing, as well as being the basis for independent preparation for the ISTQB specialist QA certification. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, practical and seminar classes, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, associate professor, professor Klymenko Iryna Anatolyivna |

|  |  |
| --- | --- |
| **Discipline** | **Methodology and organization of scientific research** |
| **Level of higher education** | First (Bachelor) |
| **Course** | 4 |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Probability theory, Algorithms and calculation methods, System programming, System software, Computer systems, Data structures and algorithms, Organization of databases, Programming, Computer architecture, Computer networks, Discrete mathematics |
| **What will be studied?** | * Methods of conducting scientific research; * Mathematical modeling methods; * Methods of displaying and transforming information. |
| **Why is it interesting/necessary to study?** | The discipline lays the foundations of independent scientific work starting with laboratory work, course and diploma design |
| **What will be learned?** | Create various models, methods of conducting scientific research |
| **How can you use acquired knowledge and skills?** | Plan scientific research, process the results of scientific research, process and analyze the obtained results |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor Selivanov Viktor Levovych |

|  |  |
| --- | --- |
| **Discipline** | **Artificial intelligence methods and technologies** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Programming, System programming, Computer architecture, Mathematics, Probability theory |
| **What will be studied?** | The main theoretical principles of the organization, construction, structures and algorithms of intelligent embedded systems and real-time systems will be studied. Acquiring practical skills in developing schedulers, quantifying the real-time data processing capabilities of computer systems and specialized processors. Laboratory works are scheduled on the following topics:   * Research and construction of correlation functions. * Applied problems of security and artificial intelligence. Neural network training. * Development of an intelligent scheduler of an embedded system. |
| **Why is it interesting/necessary to study?** | This course is designed for students who enjoy hands-on programming and problem-solving. Students will learn:   * to program in the Python language for modeling planning disciplines; * we will additionally program in Java (Kotlin) and C.   The course also covers interesting tasks from KPI-Open Olympiad programming. This will help to create effective algorithms in conditions of limited resources of embedded systems. |
| **What will be learned?** | Practical knowledge on the development of schedulers of embedded systems will be obtained. Knowledge and ability to evaluate the computing capabilities of embedded systems, based on relevant mathematical statements, and how to practically develop a scheduler using the most common scheduling methods. |
| **How can you use acquired knowledge and skills?** | After completing the course, students will be able to develop and research software and technical means for fast, including parallel, information processing in computer systems that have direct connections with real objects.  Examples of such systems are control systems for moving objects, autopilots and car safety systems, avionics devices, and virtual reality systems. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Ph.D., associate professor |

|  |  |
| --- | --- |
| **Discipline** | **Basics of data science** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Higher mathematics (mathematical analysis, linear algebra and analytical geometry), Probability theory and Mathematical statistics, Programming |
| **What will be studied?** | General concepts about the essence of data analysis, the application of the Bonferroni principle, the method of searching for nearest neighbors, document shingling. Principles of preserving the similarity of reduced sets, locally sensitive hashing and its application, distance measures. The concept of analysis of data flows of various nature and complexity and analysis of links. The market basket model and the A-Priori algorithm. |
| **Why is it interesting/necessary to study?** | Data science has significant practical implications for solving pressing everyday problems. The application of data science changes the way of life, work and thinking, since one of the conditions for the successful development of the world economy at the current stage is the ability to record and analyze accumulated data sets. Modern effective methods of working with data contribute to increasing the competitiveness of various spheres of economic activity, as well as to the specialists who possess these methods. |
| **What will be learned?** | Create models for data using generalization and feature extraction.  Use the methods of calculating the Jakarta similarity of sets.  Apply distance measures to determine similarity, including cosine distance, edit distance, Hamming distance, location-sensitive functions.  Analyze data streams by building a stream data model and filtering streams, count various elements in a stream, in particular, counting in a window and fading windows, use DGIM algorithms.  Perform link analysis using PageRank, its effective calculation, calculate thematically sensitive PageRank, determine link spam. |
| **How can you use acquired knowledge and skills?** | * Formation of initial data sets using the Bonferroni principle; * search for textually similar documents (plagiarism, mirror pages, articles from one source); * determining the similarity of online shopping and movie rating; * object recognition, comparison of fingerprints and newspaper articles; * creation of software systems for effective filtering of unwanted flow elements (incoming e-mail flows); * creation of teleport kits. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Novotarskyi Mykhailo Anatoliyovych |

|  |  |
| --- | --- |
| **Discipline** | **Statistical methods of machine learning** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Higher mathematics (mathematical analysis, linear algebra and analytical geometry), Probability theory and Mathematical statistics, Programming; skills in solving typical problems of these disciplines |
| **What will be studied?** | The main methods of machine learning for classification, clustering and regression problems (prediction), in particular, the principles of building some basic classifiers, problems that require to use statistical machine learning methods, algorithmic and computational approaches to their implementation with in-depth study of the Python programming language and using of TensorFlow libraries and THEANO. |
| **Why is it interesting/necessary to study?** | To obtain significant competitive advantages in the field of IT technologies, since machine learning is widely used in building search engines that learn to provide relevant results; models for generating targeted advertising within mass advertising campaigns; anti-spam software for filtering e-mail messages; software for detecting fraud on the Internet, in particular in the case of credit card transactions; pattern recognition and voice commands.  Machine learning is also widely used in scientific applications such as bioinformatics, medicine, and astronomy. |
| **What will be learned?** | Knowledge of the principles of constructing feature vectors, decision rules and classification; main types of classifiers; principles of building linear classifiers; principles of building nonlinear classifiers; features of the selection of features of classification and pre-processing of data.  The ability to choose the appropriate type of classifier depending on the problem to be solved; choose a set of features for classification and carry out preliminary data processing; apply algorithms for building a classifier based on a sample; perform calculations related to the training and operation of the classifier in the Python environment using the TensorFlow and THEANO libraries. |
| **How can you use acquired knowledge and skills?** | * Selection, construction, training and use of basic classifiers when solving tasks; * independent work in modern software complexes; mastering a large amount of information; * setting tasks and conducting an experiment, programming tasks for data analysis; * visualization of the obtained results. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** | Doctor of Science, professor Novotarskyi Mykhailo Anatoliyovych |

|  |  |
| --- | --- |
| **Discipline** | **Java programming language** |
| **Level of higher education** | First (Bachelor) |
| **Amount** | 4 ECTS credits |
| **Language of teaching** | Ukrainian |
| **Department** | Computer Engineering |
| **Prerequisites** | Basic knowledge of:  Higher mathematics (mathematical analysis, linear algebra and analytical geometry), Probability theory and Mathematical statistics, Programming; skills in solving typical problems of these disciplines |
| **What will be studied?** | The main methods of machine learning for classification, clustering and regression problems (prediction), in particular, the principles of building some basic classifiers, problems that require to use statistical machine learning methods, algorithmic and computational approaches to their implementation with in-depth study of the Python programming language and using of TensorFlow libraries and THEANO. |
| **Why is it interesting/necessary to study?** | To obtain significant competitive advantages in the field of IT technologies, since machine learning is widely used in building search engines that learn to provide relevant results; models for generating targeted advertising within mass advertising campaigns; anti-spam software for filtering e-mail messages; software for detecting fraud on the Internet, in particular in the case of credit card transactions; pattern recognition and voice commands.  Machine learning is also widely used in scientific applications such as bioinformatics, medicine, and astronomy. |
| **What will be learned?** | Knowledge of the principles of constructing feature vectors, decision rules and classification; main types of classifiers; principles of building linear classifiers; principles of building nonlinear classifiers; features of the selection of features of classification and pre-processing of data.  The ability to choose the appropriate type of classifier depending on the problem to be solved; choose a set of features for classification and carry out preliminary data processing; apply algorithms for building a classifier based on a sample; perform calculations related to the training and operation of the classifier in the Python environment using the TensorFlow and THEANO libraries. |
| **How can you use acquired knowledge and skills?** | * Selection, construction, training and use of basic classifiers when solving tasks; * independent work in modern software complexes; mastering a large amount of information; * setting tasks and conducting an experiment, programming tasks for data analysis; * visualization of the obtained results. |
| **Information support** | Educational and working programs of the discipline, Point-Rating System, educational and methodical complex. |
| **Form of conduct classes** | Lectures, laboratory works, independent work |
| **Semester control** | Offset |
| **Teacher** |  |

|  |  |
| --- | --- |
| **Дисципліна** | **Data analysis using the Python language** |
| **Level of higher education** | The first (bachelor) |
| **Volume** | 4 academic credits, 120 hours |
| **The language of instruction** | English |
| **Department** | Information systems and technologies |
| **Requirements to the beginning of the study** | Basic knowledge in disciplines: Programming, algorithms and data structures, mathematical analysis, discrete mathematics, probability theory. |
| **What will be studied** | Basic Python language libraries for data (NumPу, SciPy, Mаtрlоtlіb, Seaborn, Sсіkіt-lеаrn, Pаndаѕ) And their use;  Statistical processing of data in SciPy and Pаndаѕ;.  Preliminary preparation of data for analysis in Pаndаѕ and Sсіkіt-lеаrn;  Python tools for data visualization;  The basics of machine learning and the use of its basic algorithms with the help of the Sсіkіt-lеаrn library. |
| **Why it is interesting/necessary to study** | Every person in his professional activity has a business with significant volumes of data. Analysis of these data allows us to interpret them correctly, to get new information, and helps to make decisions. Python is a popular programming language that is actively used and has a wide range of data analysis tools. |
| **Why you can learn (results of training)** | To improve skills of working with Python;  Effectively use Python language libraries for data processing, visualization, and analysis;   * to choose the methods and algorithms for the preliminary data processing correctly; * correctly choose the most informative methods of data representation; * use machine learning algorithms for classification, clustering, regression analysis, etc. |
| **How to use acquired knowledge and skills (competencies)** | * efficiently process, visualize and analyze data obtained from your own experiments and studies; * create programs in Python for data. |
| **Information support** | Syllabus, lectures, laboratory works |
| **Form of instruction** | lectures, laboratory works. |
| **Semester control** | Credit, 3 course 5 semester |
| **The teacher** | Teacher Timofeeva Y.S. |

|  |  |
| --- | --- |
| **Дисципліна** | **Methods and technologies of Artificial Intelligence** |
| **Level of higher education** | The first (bachelor) |
| **Volume** | 4 academic credits, 120 hours |
| **The language of instruction** | Ukrainian, English |
| **Department** | information systems and technologies |
| **Requirements to the beginning of the study** | Basic information systems and technologies Knowledge in disciplines:. Algorithms and methods of calculation, algorithms and data structures, programming, discrete mathematics, software engineering |
| **What will be studied** | Basic knowledge about context, terms, problems and current state of research in the field of artificial intelligence. Review of basic principles and approaches to studying systems of artificial intelligence. Review of the main directions of research and corresponding tools of artificial intelligence: fuzzy systems, neural networks, genetic algorithms, evolutionary methods, deep learning, etc. Overview of basic models and algorithms with examples of their application and software-hardware means of implementation. |
| **Why it is interesting/necessary to study** | The field of artificial intelligence in recent years is experiencing an era of real rise given the rapid development of new models of neural networks, software tools and rapid development of computing resources on the basis of graphic and tensor accelerators. Research on artificial intelligence methods now shows the fastest growth in research and practical applications in many areas of life around us: From personal assistants in smartphones to self-driving cars. Development and implementation of systems with artificial intelligence allows to automate the solution of tasks that require knowledge. Tasks that were not available without the use of specialists in a certain field. |
| **Why you can learn (results of training)** | Theoretical knowledge and basic practical experience in applying different methods of artificial intelligence to existing practical problems in a wide range of applications. |
| **How to use acquired knowledge and skills (competencies)** | The acquired knowledge will allow us to understand the place and role of methods of artificial intelligence in the general context of information technologies. This discipline is a necessary stage for preparation to a professional level of mastering by specialized knowledge and skills in the field of artificial intelligence, which will be taught in the following educational disciplines, which are devoted to more detailed study of separate methods of artificial intelligence. |
| **Information support** | Syllabus, lecture notes, presentation materials, methodical instructions for laboratory work |
| **Form of instruction** | lectures, laboratory works. |
| **Semester control** | Credit, 3 course 5 semester |
| **The teacher** | Associate professor, PhD Shimkovich V.M. |

|  |  |
| --- | --- |
| **Дисципліна** | **Software tools for designing and implementing neural network systems** |
| **Level of higher education** | The first (bachelor) |
| **Volume** | 4 academic credits, 120 hours |
| **The language of instruction** | Ukrainian |
| **Department** | information systems and technologies |
| **Requirements to the beginning of the study** | Basic knowledge in disciplines: Intellectual systems and technologies,. Programming in the Python language |
| **What will be studied** | Design, realization and research of software systems using artificial intelligence. Methods of creation, training and functioning of neural network systems and deep learning systems. Software tools for realization and research of neural network systems on Python using the libraries TensorFlow and Keras. |
| **Why it is interesting/necessary to study** | Machine learning is a section of computer science that studies the development of algorithms that can be taught. One of the sections of machine learning is artificial neural networks. Deep learning is one of the hottest areas of data science, as many thematic studies yield impressive results in robotics, image recognition, and artificial intelligence.  One of the most powerful and easy-to-use Python libraries for the development and evaluation of deep learning program models is Keras. It consists of efficient number libraries for Theano and TensorFlow. Studying data of program technologies of realization and research of neural network systems will be easy and simple. implement complex software systems of artificial intelligence. |
| **Why you can learn (results of training)** | In studying this discipline, students learn theoretical fundamentals of neural networks and get initial experience in software development, which implements neural network technologies.  At the laboratory lessons will receive initial experience of creation of software systems on the basis of neural networks. Develop and research neuro networking systems of different purposes using the Payton programming language and TensorFlow and Keras libraries. |
| **How to use acquired knowledge and skills (competencies)** | Acquired knowledge can be used in the development of software systems with artificial intelligence. |
| **Information support** | Syllabus, lecture notes, presentation materials, methodical instructions for laboratory work |
| **Form of instruction** | lectures, laboratory works. |
| **Semester control** | Credit, 3 course 5 semester |
| **The teacher** | Associate professor, PhD Shimkovich V.M. |