Field of Study: Computer science

Offered courses for the academic year 2021/2022, summer semester, with descriptions

Semester 2

Object - Oriented Programming

The course aims to:

- develop knowledge about the paradigm of object programming,
- understanding the principles and logic of writing object-oriented applications,
- understanding the benefits of programming and object design

During the classes, students will learn the principles of creating applications in Java using object oriented paradigms.

Topics:

- NetBeans environment. Variables.
- Familiarization with input/output operations.
- Classes, objects, fields and methods.
- Inheritance
- Interfaces and abstract classes
- Polymorphism
- Java Swing
- Graphics in Java
- Traffic animation
- Algorithms in Java:
  - Hanoi towers
  - Fractals
  - Merge sort

Digital Electronic

The digital electronic has the task to provide students with basic knowledge and skills regarding design, research, diagnosing the correctness of operation and removal of defects of digital circuits.

After completing the subject:

- student should have a basic knowledge of counting systems,
- student should know basic logical functions and the rules of Boolean algebra,
- student should be able to determine the minimal notation of a logical function using function minimization logical arrays by Karnaugh's method and Quin-McQluskey's method,
- student should be able to design complex, multi-input digital circuits,
- student should be able to use computer software for design and simulation operation of digital circuits,
- student should be able to design asynchronous digital circuits,
- student should be able to design synchronous digital circuits,
- student should be able to design sequential digital circuits,
- student should be able to perform basic measurements in digital circuits,
- student should diagnose and repair faults in digital circuits,
- student should be able to create simple PLC programs,

Programming Technologies

The Programming Technologies course covers all the basics of programming in Python, as well as general computer programming concepts and techniques. The course also familiarizes the student with an object-oriented approach. Students who complete the course will be able to accomplish coding tasks related to the basics of programming in the Python language and understand the fundamental notions and techniques used in object-oriented programming.

Topics:

- Installation of the environment
- Introduction to Python
- Formatting strings
- Conditional instructions
- Functions
- Classes and objects
- Dictionaries
- Lists
- Practical tasks

Introduction to Computer Networks

The course introduces participants to computer networks. The semester covers topics related to networking terminology and protocols, local area networks (LAN) and wide area networks (WAN), OSI (Open System Interconnection) model, cabling, cabling tools, Ethernet technology, IP (Internet Protocol) addressing and standards for computer networks. The student learns the functions of well-known TCP/IP applications (e.g. WWW, e-mail) and related services (HTTP, DNS, SMB, DHCP, SMTP/POP, and Telnet) and the services, technologies, and problems that professionals encounter when designing, building, and managing modern networks. Student identifies key network components that transmit data and learns the characteristics of network architectures such as: fault tolerance, scalability, quality of service and security. Student is able to solve basic communication problems on a local network.

Topics:

- Initialization of switches and routers
- Building a simple network.
- Analysis of network standards, preview of network traffic
- Ethernet cabling.
- Ethernet frames.
- Building a network with a switch and a router.
- TCP/UDP frame analysis.
- IPv4 and IPv6 addressing.
- Dividing the network into subnets.
- Analysis and performance tracking of application layer protocols.
- Creating a small computer network.

Semester 4

Software Engineering

This course is an introduction to the practical problems of specifying, designing, building and delivering reliable software systems. Students learn how to describe and design systems using software engineering tools. Presented topics refer to the practical issues in software engineering. During the classes, students create the IT system project, using the software engineering method for its description.

Topics:

- Steps in the software development process
- Use cases. Recognizing the actors. Relationships between use cases.
- Activity diagram - features. Notation. Control flow. Decisions.
- Class diagram - features: Class symbol, notation, access levels, static components, tuples. Relationships between classes.
- Objects and associations.
- Participants on the sequence diagram. Message types. Blocks. Interaction operators. Actors.

- Syntax. Operations and creating a communication diagram.
- Advanced components of a time diagram. Events and time constraints. Alternative ways of presenting states. Message transfer.
- Parts of an overview interaction diagram.

Internet of Things Technologies

In this course, students will explore various components of Internet of Things such as sensors, peripheral, networking and client- server communication. After successful completion of this course, students will be able to:

- understand general concepts of Internet of Things,
- recognize a variety of devices, sensors, and applications,
- analyze various M2M and IoT architectures,
- create IoT solutions using sensors, actuators and devices,

Topics:

- Configuration, operation of IOT devices, peripheral controls (WiFi, GPIO, ADC, servomechanisms).
- Connecting devices to the Internet (eg. devices responsive to the current weather forecast).
- Exposition of the functionality of devices in the form of websites (1) (SOAP, REST).
- Exposition of the functionality of devices in the form of services (2) (COAP).
- M2M (Machine-2-Machine) information (broadcast communication protocols).
- M2M (Machine-2-Machine) information (Communication based on the exchange of messages eg MQTT, MQTT-S).
- Distributed data processing from IOT (Complex Event Processing).

Computer Graphics

The subject of the course is related to the presentation of the methods of designing raster and vector graphics using graphic applications and learning about programming methods of operations in 2D and 3D space in the selected programming language.

Topics:

- Graphics in office programs
- Working with Corel family programs
- Working with Adobe software and/or Blender
- Implementation of raster algorithms in a selected programming language
- Implementation of algorithms for operations in 2D space in the selected programming language

- Implementation of algorithms for operations in 3D space in the selected programming language
- Practical application of the OpenGL library
- computer animation creation

Internet Application Programming

The aim of the course is to familiarize students with the internet application design process. The course provides a discussion of network protocols whose understanding is necessary for the correct design of web applications. It is an introduction to creating applications on the client's side and server side, taking into account current technologies and security issues in a web environment.

Topics:

- HTTP protocol in action: practice with web browser programming tools.
- Creating HTML documents. Using structural elements, references, tables, text formatting and attaching multimedia.
- Creating Forms in HTML.
- Exercises in using cascading style sheets. Applying CSS rules for text, blocks and positioning of elements on the page.
- Javascript programming. Event handling. Interacting with an HTML document using the Document Object Model (DOM). Communication with the server in the background (AJAX).
- Creating WWW pages using Bootstrap library.
- Project: Preparation of a complete website using HTML, CSS, Javascript.
- Exercises in the basics of WWW server configuration and management.
- Programming in PHP language. HTML document generation. Handling data from forms. File System Interaction. Communication with the database.

Programming Microcontrollers

The main objectives of the course are presentation of the scope from the basics of construction and programming of microcontrollers, learning to create simple and complex electronic circuits controlled using a microcontroller and the practical use of the microcontroller in everyday life. Classes take place on the basis of the Arduino family of microcontrollers.

Topics:

- Preparation of microcontroller device. Installation of the environment. Uploading the first program.
- UART communication. Serial transmission. Global and local variables in Arduino.
- ADC. Servos. PWM. Voltage measurement by means of an A/D converter.

- Arduino external libraries.
- Servos.
- Control of DC motors.
- LCD Displays. I2C communication.
- Sensors. Measuring humidity, distance, temperature.
- Create charts with the Arduino and its built-in tool.
- Interruptions. Arduino interrupt handling.

Graphical User Interface Design

The purpose of this lecture is to introduce students to the basic theoretical and practical issues of implementing a graphical user interface. In particular, familiarize them with terminology and selected technologies, (WPF). During the classes, the Visual Studio environment and the C# programming language are used.

Topics:

- Overview of basic WPF controls, basic layout managers: StackPanel, WrapPanel, Grid. Working with XAML files.
- Examples of using property bindings between controls using XAML
- Binding data between the logic and presentation layers of the application, DataContext, ItemsSource
- Implementation of event handling methods
- Simple GUI user interface design, debugging, error handling,
- Design of an application of medium complexity using various GUI controls,. Using standardized dialog boxes.
- Designing custom controls, RoutedEvents mechanism of WPF library
- Advanced layout managers, linking containers like Grid, StackPanel, WrapPanel, DockPanel
- Converters, Conversion of values with IValueConverter
- Debugging data binding, output window, setting TraceLevel, using debugger
- Example of application using commands interface
- Exception handling in WPF
- Working with CultureInfo objects, properties of the Application class: Culture / UICulture
- Application using DispatcherTimer, BackgroundWorker