

**RTU Course "Programming Languages"**

12308 null

General data

Code	DIP122
Course title	Programming Languages
Course status in the programme	Compulsory/Courses of Limited Choice
Course level	Undergraduate Studies
Course type	Professional
Field of study	Computer Science
Responsible instructor	Larisa Zaiceva
Academic staff	Jeļena Jevsjukova Aleksejs Jurenoks Inese Šimkeviča Normunds Kante
Volume of the course: parts and credits points	2 parts, 4.0 Credit Points, 6.0 ECTS credits
Language of instruction	LV, EN
Annotation	Programming language concept, classification and usage possibilities in project development. Classification of programs and methods for software development. Language syntax and semantics. Basic programming languages, operations with data, management operator overview. Functions, Object scope and memory classes. Characteristics, arrays, character string processing. Structures. Working with files. Graphic tools and objects.
Goals and objectives of the course in terms of competences and skills	The aim of the course is to introduce students to different programming languages, programming language design concepts, and demonstrate software development and implementation using PHP, JavaScript, Python language. Tasks: 1) to observe programming languages different classes, analyzing their syntax and usage possibilities; 2) to consider programming language, PHP, JavaScript and Python concepts and structure, their syntax and semantics; 3) to teach students to develop and implement programs for solving various tasks; 4) to teach students to use and to apply in practice the up-to-date library and frameworks.
Structure and tasks of independent studies	Execution of laboratory works: 1) programming of branched processes; 2) mass processing with functions and procedures; 3) processing of symbols sequences; 4) programming of the branched processes; 5) processing of symbol strings; 6) processing of information using the SQL language; 7) web software integration.
Recommended literature	1. Programming PHP. – O'Reilly and Associates; Vol:3 (2013). ISBN: 1449392776. 2. Python for Data Analysis - O'Reilly; 2nd ed. edition (2017). ISBN: 1491957662 3. Learning JavaScript Data Structures and Algorithms – Packt Publishing (2014). ISBN: 1783554878 4. Learning PHP, MySQL & JavaScript. - O'Reilly; 5th ed. edition (2018). ISBN: 1491978910 5. Data Structure and Algorithmic Thinking with Python: Data Structure and Algorithmic Puzzles. - CareerMonk Publications (2015). ISBN: 8192107590
Course prerequisites	Secondary school programme

Course outline

Theme	Hours
The concept of programming. Classification of programming languages.	4
Programming language PHP core elements: data types, variables, operators (input, output, loop).	14
JavaScript language basic elements and structure: data types, variables, operators (input output, condition, loop).	10
Programming languages basic elements of Python: data types, variables, operators (input, output, cycle).	14
Process automation methods and libraries.	4
Processing information using SQL language between platform applications.	9
File structures, data processing, data classification.	5
Network components and libraries. Client/Server software integration	4

Learning outcomes and assessment

Learning outcomes	Assessment methods
Student knows the programming languages, their classification, usage possibilities, the basic elements of programming languages.	Knowledge validation using quiz tasks
Student is able to use programming language syntax and semantics, the application of different languages, programming technologies, and key elements of the language.	Positive assessment for independent laboratory work.
Student is able to develop loop based applications	Positive assessment for independent laboratory work.
Student is able to develop application for array processing.	Positive assessment for independent laboratory work.

Student is able to develop application for string processing.	Positive assessment for independent laboratory work.
Student is able to implement logical structure and sub function.	Positive assessment for independent laboratory work.
Student is able to use automation tools and libraries.	Positive assessment for independent laboratory work.
Student is able to use external data for information processing.	Positive assessment for independent laboratory work.
Student is able to demonstrate theoretical and practical knowledge about the use of programming languages for solving practical problems.	Examination

Study subject structure

Part	CP	Hours per Week			Tests		
		Lectures	Practical	Lab.	Test	Exam	Work
1.	2.0	1.0	0.0	1.0		*	
2.	2.0	1.0	0.0	1.0		*	