RTU Course "Programming Language C ++ for Controllers Management"

33000 Faculty of Computer Science, Information Technology and Energy

General data

Code	DIP256
Course title	Programming Language C ++ for Controllers Management
Course status in the programme	Compulsory/Courses of Limited Choice; Courses of Free Choice
Responsible instructor	Aleksejs Jurenoks
Volume of the course: parts and credits points	1 part, 6.0 credits
Language of instruction	LV, EN
Annotation	The study course examines the C / C ++ programming language architecture and its implementation in the context of electronics management. The theoretical review of existing libraries and development tools (compilers) is offered in the context of electronics (robotics) management. The interaction between external input / output devices components and management methods, process automation methods are considered. In the context of the Raspberry Pi system, features, procedures, work with arrays, symbol strings, text files and objects are considered. Using the connected input / output devices, practically the data flow control and control methods are discussed.
Goals and objectives of the course in terms of competences and skills	The aim of the course is to introduce students to the concept of developing the C / C ++ programs and its application in management and control using Raspberry Pi system. The tasks are to learn C ++ language applications in the context of robot management; mastering work with sensors and output / output devices; learn to choose for the specified task appropriate programming scenario; to master the basics of program automation and the basics of remote process management, using Client / Server technology exchange of data between programs and systems, process sharing opportunities. After completing the course, students will have the skills and competences for professional use of the C ++ progra
Structure and tasks of independent studies	During the lecture the lecturer presents the theoretical concepts to the students and their practical implementation tools. At the end of each lecture, the teaching staff gives students the next task for practical realization. Students start practicing the task at the closest laboratory time and if they fail to complete a computer class, they will complete their own work. In this case, an independent work must be defended during the next laboratory work. Students who miss the defense within the set time should defend their completed consultation work.
Recommended literature	Obligātā/ Obligatory 1. Derek Molloy. Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux Wiley; 1 edition (3 Jun. 2016) 2. James Strickland . See this image Raspberry Pi for Arduino Users: Building IoT and Network Applications and Devices Apress; 1st ed. edition (10 Aug. 2017) 3. Kimmo Karvinen . Make: Getting Started with Sensors: Measure the World with Electronics, Arduino, and Raspberry Pi Maker Media, Inc; 1 edition (24 Aug. 2014) Papildu / Additional 4. Phil Gardner. Learn to Program Using C++ on the Raspberry Pi: An easy introduction to programming for beginners using Linux and GNU C++ Nielsen; Revised edition edition (28 Aug. 2017) 5. Warren Gay. Exploring the Raspberry Pi 2 with C++ Apress; 1st ed. edition (24 Nov. 2015)
Course prerequisites	Practical skills in application development in high level programming languages.

Course contents

Content	Full- and part-time intramural studies		Part time extramural studies	
	Contact Hours	Indep. work	Contact Hours	Indep. work
C ++ language compilers and their applications	4	6	0	0
Raspberry Pi architecture	6	9	0	0
Imperative programming languages. Program structure, operations with data, management structures	4	6	0	0
Data Types. Primitive data types. Work with indicators and arrays, processing of symbol strings. Structures	6	9	0	0
Receiving information from sensors / sensor properties, working with streams and files	12	18	0	0
Using the C ++ language library for system management	10	15	0	0
Actuator control (motors / pneumatics)	10	15	0	0
Sub-programs. Object scope and memory classes	6	9	0	0
Use of Client / Server technology in process management	6	9	0	0
Total:	64	96	0	0

Learning outcomes and assessment

Learning outcomes	Assessment methods
The student knows the basic elements of the C ++ programming language, the language features and possibilities of use, their classification according to different parameters, the basic structure of the language and their use in the creation of programs.	Successful completion of the test and practical tasks
The student is able to develop programs using imperative programming language, which provides processing of information with array, symbolic strings, indicators, structures and files.	Successful completion of the test and practical tasks
The student is able to receive information from sensors, knowledge of sensor properties, is able to manage data flows and provide executive control.	Successful completion of the test and practical tasks
Student is able to develop and practice Client / Server technology for process management.	Successful completion of the test and practical tasks
The student is able to demonstrate theoretical and practical knowledge of C ++ language usage possibilities, is able to process the information received from sensors and to manage executive mechanisms.	Successfully passed the exam.

Study subject structure

Part	CP	Hours			Tests			Tests (free choice)		
		Lectures	Practical	Lab.	Test	Exam	Work	Test	Exam	Work
1.	6.0	2.0	0.0	2.0		*		*		