RTU Course "Computer Studies (basic course)"

13223 Department of Fundamentals of Electronics

General data

Code	RTR105
Course title	Computer Studies (basic course)
Course status in the programme	Compulsory/Courses of Limited Choice
Responsible instructor	Artūrs Āboltiņš
Academic staff	Viktors Zagorskis Tatjana Solovjova
Volume of the course: parts and credits points	1 part, 3.0 Credit Points, 4.5 ECTS credits
Language of instruction	LV, EN
Annotation	The study course aimed at the first-year bachelor students provides insights into contemporary computers, programming languages frequently used in electronics and telecommunications, helpful integrated development environments and elementary computing algorithms to be applied in further studies and engineering work. The study course outlines mathematical, technical and philosophic principles of data acquisition, computing and representation systems, which provide the basis for the development of professional expertise and practical competence working with professional computer systems based on open-source operating environment Linux – UBUNTU implementation in particular. The students get acquainted with classical programming languages C, C++ and contemporary programming language Python. Binary and hexadecimal numeral systems are studied. The concepts of ‘bit’, ‘byte’ and ‘data type’ are comprehensively analysed. Practical programming tasks are related to simple tasks completed within the framework of the courses in physics, electrical and mechanical engineering, as well as to a range of themes in advanced mathematics such as simple function differentiation and area computing of functionally developed geometric figures. The tasks connected to complex numbers, bit operations and data sorting are completed depending on the specifics of the study program.
Goals and objectives of the course in terms of competences and skills	The goal of the study course is to provide an overview of C, C++ and Python programming languages and free access to technical tools for the development of simple but professional software. Objectives of the study course: 1. To introduce modern, professional and efficient programming environments based on Linux operating system, with a strong focus on the command-line interface. 2. Give an opportunity to gain practical experience in designing simple C++ codes and Python scripts by algorithmizing simple physics and math problems. 3. To develop procedural programming skills. 4. To teach the simplest approaches of data acquisition, processing and representation.
Structure and tasks of independent studies	Students are studying study course literature, they independently solve pieces of homework, independently get prepared for laboratory works and create laboratory works’ reports, get prepared for the exam.
Recommended literature	Obligātā/Obligatory: [1] Ziemelis J. Ievads algoritmu valodā C. Rīga, RTU, 2006. Papildu/Additional: [2] Stroustrup, B. The Programming Language C++. Addison-Wesley, 2009. [3] The C++ Resources Network. Available: http://www.cplusplus.com/ [4] Elkner, J., Downey, A.B., Meyers, C. How to Think Like a Computer Scientist, Learning with Python. 2nd ed. 2008. Available: http://www.openbookproject.net/thinkcs/python/english2e/ [5] Elkner, J., Downey, A.B., Meyers, C., McCane, B., Hewson, I., Meek, N. Practical Programming in Python. 2009. Available: http://www.cs.otago.ac.nz/staffpriv/mccane/Downloads/PracticalProgramming.pdf [6] Knuth, D. All questions answered. Notices of the American Mathematical Society, vol. 49(3), 2002, p.318–324. [67] Herrmann, D. C++ für Naturwissenschaftler: beispielorientierte Einführung. Addison-Wesley, 2001. [8] Shader, M., Kuhlins, S. Programmieren in C++: Einführung in den Sprachstandard. Springer, 1998. [9] Dausmann, M., Broeckl, U., Goll, J. C als erste programmiersprache.Teubner, 2005. [10] Zagorskis, V. Datormācība-pamatkurss. Laboratorijas darbu apraksti. Rīga: RTU, ETF, SC, 2010. [11] Zagorskis, V. Datormācības lasījumi. I. daļa. Rīga: RTU, ETF, SC, 2009.
Course prerequisites	High school education level knowledge of subjects in mathematics and use of computer.

Course contents

Content	Full- and part-time intramural studies		Part time extramural studies	
	Contact Hours	Indep. work	Contact Hours	Indep. work
Human and Computer. Similarities and differences. Interoperability philosophy. Information. Information unit. Bit. Byte	1	1	0	0

Operational systems (OS). Why OS systems are needed? Concept. History. How OSes are developed?	1	1	0	0
UNIX. Multiuser and multiprocess system. Time divided processes. Linux.	1	1	0	0
OS kernel. Shell. Physical devices. File systems.	1	1	0	0
Users in Linux. Terminal application. Shells. Shell commands. Variables, directories, files. Rights.	2	2	0	0
Data-In. Data-Processing. Data-Out. Programming in shell. Shell scripts.	2	2	0	0
C, C++ languages. Principles of languages. Syntax. Building blocks. Operators. Variables. Statements.	2	2	0	0
Main function in C, C++. How to compile C, C++ program? Pre-processing directives.	2	2	0	0
How to execute a binary file? Data evaluation. Data formatting. Standard output. Error output.	2	2	0	0
C, C++ libraries. Data type - char. Declaration and definition of variables.	2	2	0	0
Circle of numbers. Natural numbers and integers. Characters. ASCII. Arithmetical operations. Priorities.	1	1	0	0
Data exchange. Algorithms. Alg I. Dec to Bin conversion.	2	2	0	0
Different Integers. Short. Int. long data types. Library file limits.	2	2	0	0
Floating point numbers. Float, double, long double data types. Data arrays.	2	2	0	0
User defined C, C++ functions. Prototypes. Program block.	2	2	0	0
Looping with FOR and WHILE. Loop variable. Loop body.	2	2	0	0
Alg II. Factorial. Alg III. Bin to Dec conversion.	2	2	0	0
Some useful functions from STD - standard C library.	1	1	0	0
Data type - BOOLEAN. Program branching. Logical operations. Bitwise operations.	2	2	0	0
Data arrays. Pointers. Pointers arithmetic. References. User defined data types: structures, etc..	2	2	0	0
How to format data and write to file? How to read the file? Text and binary files.	2	2	0	0
Python programming language elements. Arithmetical operations. Data-In. Data-Out. Data types.	2	2	0	0
Data types in Python. Integers, floating point numbers, complex numbers.	2	2	0	0
Advanced data types: list, dictionary. Looping in Python. How to read data lines from file?	2	2	0	0
Alg. iV. Factorial. Alg. V. Number series. Recurrence.	2	2	0	0
Python standard libraries (sys, os, math, cmath, numpy, etc.). User functions. Lambda function.	2	2	0	0
Alg. VI. Finding of math-function root.	2	2	0	0
Branching in Python. IF. IFELSE. ELSE. Logical operations. Bitwise operations.	2	2	0	0
Alg. VII. Data sorting. Bubble algorithm.	2	2	0	0
Alg. VIII. Function differentiation.	2	2	0	0
Alg IX. Finding of function integral.	2	2	0	0
Scientific libraries for Python. SciPy. NumPy. Matplotlib. Independent tool GNUPLOT.	2	2	0	0
User programs for scientific usage (Octave, Maxima, etc.).	1	1	0	0
Elementary descriptive statistics.	1	1	0	0
Total:	60	60	0	0

Learning outcomes and assessment

Learning outcomes	Assessment methods
Is able to use professional UNIX like operational systems in a terminal environment. Have the practical experience to manipulate directories, files, work with text editors, invoke scripts and programs.	Pieces of homework scored. Tests. Exam.
Is able to write simple C, C++ programs based on looping, branching and conditioning procedures. Knows C, C++ program compiling, syntax error finding, executing, terminal output redirection.	Programming tasks scored. Labs scored. Tests. Exam.
Can create simple programs in Python based on some fundamental algorithms like function root finding, definitive integral value calculation, function differentiation, simple bubble sorting.	Programming tasks scored. Labs scored. Tests. Exam.
Is able to operate GNUPLOT as math-functions 2D visualisation and evaluation tool.	Labs scored. Exam.

Evaluation criteria of study results

Criterion	%
Attendance	10
Tests	5
Programming tasks	25
Homework	10
Lab work (codes)	10
Lab work (defence)	15
Exam	25
Total:	100

Study subject structure

Part	CP	Hours per Week			Tests		
		Lectures	Practical	Lab.	Test	Exam	Work
1.	3.0	1.5	0.0	1.5		*	