RTU Course "Computer Studies (special course)"

13223 Elektronikas pamatu katedra

General data

Code	RTR108
Course title	Computer Studies (special course)
Course status in the programme	Compulsory/Courses of Limited Choice
Course level	Undergraduate Studies
Course type	Academic
Field of study	Electronics and Telecommunications
Responsible instructor	Viktors Zagorskis
Academic staff	Dmitrijs Puriševs Tatjana Solovjova Jurijs Ivanovs Jānis Klūga Māris Tērauds
Volume of the course: parts and credits points	1 part, 2.0 Credit Points, 3.0 ECTS credits
Language of instruction	LV, EN, RU, DE
Possibility of distance learning	Not planned
Abstract	The course provides further insights into contemporary computers and computing algorithms to be applied in further studies and engineering work, the basics of which were acquired within the framework of the course Computer Studies (basic course) taught during the first semester at RTU FET bachelor programme. The course outlines general data acquisition, computing and representation methods, related to potential automatization of these processes, using Internet recourses (telecommunication channels and Internet sites). Students get acquainted with Python programming language, study HTML elements presenting the data, and XML elements structuring the data. Within the framework of the course the students create electronic documents with the help of document production system LaTex. C++ and Python programming language object oriented programming elements are studied. The working computer environment is Linux based operational system like SUSE and/or UBUNTU. Practical programming tasks are connected with the special requirements imposed by the study field of electronics and telecommunications.
Goals and objectives of the course in terms of competences and skills	The goal of this course is to provide an overview of programming languages and free access tools available to solve simple computational problems in a specialised field of electronics and telecommunications. Students acquire skills, practical experience and technical competences in data acquisition, processing and presentation.
Structure and tasks of independent studies	There are seven groups of assignments. Student has to be prepared for each of them. I. Lectures. Every week starting from the first. 8 lectures in total. II. Home works. Every week starting from the first. 8 home works in total. III. Labs. Every second week starting from the first. 8 labs in total. IV. Work on video presentation. Weeks 9-16. The student individually develops a video presentation on the selected theme using the tolls, algorithms and methods covered during the semester. V. Midterm interactive test. Week 9. VI. Final interactive test. Week 16. VII. Exam. Duration 4 hours. Oral with practical problem set.
Recommended literature	[1] Zagorskis, V. Datormācība-spectkurss. Praktisko darbu apraksti. Rīga: RTU, ETF, SC, 2010. Available: http://213.175.92.39 [2] Course RTR108 class notes. Available: http://213.175.92.39 [3] Python documentation. Available: http://docs.python.org/ [4] Scientific Python. Available: http://openbookproject.net//thinkCSpy/ [5] Scientific Tools for Python. Available:http://www.scipy.org/ [6] Intro to programming with Python and Tkinter. Available:http://wiki.python.org/moin/ [7] Dalheimer, K. LaTeX - kurz & gut, 2. Auflage. O'Reilly, 2004 [8] Oetiker, T. Not so short introduction to LaTeX2e. Available: www.ctan.org/tex-archive/info/lshort/english/lshort.pdf [9] Online Web building tutorials. Available: http://www.w3schools.com/
Course prerequisites	1. Linux user experience.2. C, C++ beginner skills. Knowledge of Python elements desired.3. Skills in elementary algorithm development for technical tasks.4. Ability to read technical texts in Latvian, Russian, English and German texts.

Course outline

Theme	Hours
Data types and specifications of data exchange in Electronics and Telecommunications.	2
Human (engineer) tools for data acquisition, processing and presentation.	2
HTML language - standard tool for data presentation in the Internet environment.	2

URL and WEB resources. Data exchange using networking. How to get data automatically.	2
Data structuring using XML standard. Document Object Model (DOM). Data parsing.	2
Object oriented programming (OOP) vs. procedure programming. First approach toward OOP in C++ and Python.	3
Data visualisation techniques on user displays. Pixels. Major graphic file formats.	3
Modular programming. Large projects. How are these developed?	2
Linux shell. Command line arguments. Shell scripts. How to manage data automatically.	2
Scientific computing using Python. Numerical data visualisation. Tkinter - graphical user interface (GUI) package.	12

Learning outcomes and assessment

Learning outcomes	Assessment methods
Ability to write a well formatted simple scientific specified documents using LaTex documentation environment. Outcomes are various PDF documents ready to be published electronically.	8 Home works, scored. 4-hour exam
Knowledge to choose useful programming tool or language or library to solve data processing problem for carrying out data visualisation as graphical file(s) or form based applications.	2 Labs, scored. 4-hour exam
Skills in using advanced level programming language Python for scientific 6-Labs,specified computation by searching, getting and processing data from networking resources.	2 Labs, scored. 4-hour exam
Ability to get optimal abstraction level to develop object oriented programs in Python in order to solve elementary electronic circuits.	2 Labs, scored.. 4-hour exam
Student is able to create, to develop and support simple, in HTML written WEB pages. Skills to compute and include XML based SVG drawings.	2 Labs, scored. 4-hour exam
Video based presentation that demonstrates personal, intellectual and creative characteristic features of the student.	Course work only (9-16 weeks), scored.

Study subject structure

Part	CP	ECTS	Hours per Week			Tests		
			Lectures	Practical	Lab.	Test	Exam	Work
1.	2.0	3.0	1.0	0.0	1.0		*	