POLITEHNICA University of Bucharest
Faculty of Electronics, Telecommunications and Information Technology

LECTURES DESCRIPTION

1. Program identification information

1.1 Academic institution	University POLITEHNICA in Bucharest
1.2 Faculty	Electronics, Telecommunications and Information Technology
1.3 Department	Dept. of Applied Electronics and Information Engineering
1.4 Domain of studies	Electronic Engineering, Telecommunications and Informational Technologies
1.5 Cycle of studies	Licence
1.6 Program of studies/Qualifications	Technologies and Systems of Telecommunications

2. Course identification information

2.1 Name of the course				**Numerical Methods**			
2.2 Lecturer				Ş.l. dr. ing. GROSU Vlad-Alexandru			
2.3 Instructor for practical activities				Ş.l. dr. ing. GROSU Vlad-Alexandru			
2.4 Year of study	II	2.5 Semester	3	2.6 Evaluation type	Continuous assessment	2.7 Course choice type	Mandatory

3. Total estimated study time (hours per semester for academic activities)

3.1 Number of hours per week	2	out of which: 3.2 lectures	1	3.3 practical activities	1
3.4 Total hours in the curricula	28	out of which: 3.5 lectures	14	3.6 practical activities	14
Distribution of time:					hours
Study based on: official manual, lecture notes, bibliography and hand notes					16
Supplementary study materials (library, electronic access guidelines, in the field etc.)					6
Seminars/laboratories preparation, homeworks, assessments, portfolios and essays					10
Tutoring					1
Examinations					3
Other activities					0

3.7 Total hours of individual study	36
3.9 Total hours per semester	78
3.10 Number of ECTS credit points	3

4. Prerequisites (where applicable)

4.1 Curricular	• Algebra and Mathematical Analysis • Computer programming • Algorithms and data structures
4.2 Competence based	• Able to apply the knowledge, concepts and fundamental methods in Algebra and Mathematical Analysis • Identify and solve practical situation involving algorithms and data structure elements, microprocessor/microcontroller specific programming techniques and domestic usage.

5. Requisites (where applicable)

5.1 Lecture related (teaching activities)	Video projector required, along with all related accessories (e.g. power cable, data and signal cables, remote control).
5.2 Applications related (seminar/laboratory)	Mandatory attendance (according to UPB's inner active Regulations)

6. Specific competences (cumulated)

Professional abilities	C1. Using of fundamental elements that refer to the electronic devices, circuits and instrumentation C2. Application, in typical situations, of basic methods of signal acquisition and processing C3: Ability to correctly apply the knowledge, concepts and fundamental methods with respect to computational systems' architecture, microprocessors, microcontrollers, programming languages and techniques..
Transversal abilities	It is not the case

7. Course objectives (concluded from the grid of cumulated specific abilities)

7.1 General objective of the course	– **Lecture**: • To understand the programming field's lingo; • To provide a comprehensive understanding of the fundamental concepts that the typical numerical methods used in Electronic applications and/or design rely on.; • Acquaintance with ANSI C programming language; • Develop algorithmic thinking and use it throughout the analysis of the studied algorithms. – **Laboratory**: • Develop general computational routines used in typical applications for programming field. Use C programming language in design and implementation of programs thus ensuring their portability and effective usage as well as their interface with the large majority of development/design software packages based on C language. • Present the algorithms in pseudo-code that facilitates their implementation in a high-level programming language, whatever it would be. • The complete programs developed during the laboratory workshops become useful tools for the students, not merely in their annual projects activities or License Degree but also for their future engineer careers.
7.2 Specific objectives	Develop skills and correctly apply the structured programming paradigms when implementing personal programs/applications. Exercise these skills in identification of the problems' requirements and generalize their solutions where necessary. Choose the proper solution that offers a correct resolution for all the problem's practical situations (so called the problem's universe). Understand the peculiarities of the chosen solving algorithms. Develop and exercise the ability to compare between various algorithms for a certain problem and therefore choose the most suitable. Think about optimized approaches for a variety of real life situations.

8. Content

8.1 Lecture	Teaching techniques	Remarks
Introduction. Absolute and relative errors. The IEEE-754 floating point standard. Errors' classification of errors in typical numerical environments. Procedural graphs.	Interact with the students throughout the set of problems linked to the teaching methods. There are parts of the lecture reserved for presentation and resolution of current specific problems. The modeling part often translates to the announcement of the solving principles of typical programming problems which require immediate results. The continuous dialogue throughout the lecture extends during the tutoring meetings as well. These meetings are necessary for students' preparation required, both by the laboratory's final test and the final examination.	3 hours
Numerical algorithms for solving algebraic equations. Methods used in determination of real solutions for both polynomial equations and transcendental equations.		2 hours
Linear and non-linear equation systems of equations. Direct and indirect methods.		4 hours
Numerical differentiation algorithms. Symmetrical and asymmetrical approaches in computing 1st order derivative of a function. Higher order differentiation techniques.		2 hours
Numerical integration algorithms. Quadrature and cubature methods: simple and double integrals. Improper integrals.		3 hours
Methods used in functions approximation. Polynomial interpolation.		3 hours
Methods used in functions approximation. Regression (or optimization) methods based on the least squares method.		2 hours
Differential equations and systems. Euler and Runge-Kutta classes of methods. Comparison between various algorithms.		3 hours
Numerical resolution of the integral equations.		2 hours
Eigenvalues and vectors of eigenvalues.		2 hours
Special functions.		2 hours

Bibliography

- I. Rusu, Dana Gavrilescu, Vlad Al. Grosu - "*Programarea calculatoarelor în limbaj C*", Editura MatrixRom, Bucureşti, 2002.
- I. Rusu, Dana Gavrilescu, Vlad Al. Grosu – "*Îndrumar de laborator pentru programarea calculatoarelor: C*", Editura MatrixRom, Bucureşti, 2004.
- I. Rusu, Vlad Al. Grosu – "*Programarea calculatoarelor în limbaj C: probleme rezolvate şi comentate*", Editura MatrixRom, Bucureşti, 2008.
- D.I. Năstac, "*Programarea calculatoarelor în limbajul C – Elemente fundamentale*", Editura Printech, Bucureşti, 2006.
- D.I. Năstac, "*Structuri de date ţi algoritmi – Aplicaţii*", Editura Printech, București, 2008.

<table>
<tr><td colspan="3">

- D. Burileanu, C. Dan, M. Pădure, "*Programare în C. Culegere de probleme*", Editura Printech, Bucureşti, 2004.
- Brian Kernighan, Dennis Richie – "*The C programming language*", Prentice Hall, New Jersey, ediţiile 1978 şi 1988.

</td></tr>
<tr><td>8.2 Laboratory</td><td>Teaching techniques</td><td>Remarks</td></tr>
<tr><td>Numerical algorithms for solving algebraic equations.</td><td rowspan="8">In the first place a short review of the theoretical approaches on the subject is given.
The goal is for the students to write fully functional programs based on the provided pseudo-code in the guidelines.
The programming language is ANSI C (according to either -std=c99 or –std=c11)

Practical problems are fully presented. This approach offer the possibility to perform comparative evaluations and to take the optimum decisions when several algorithms are available for a given problem, so that one could choose the most suitable program in a real situation.</td><td>2 hours</td></tr>
<tr><td>Linear and non-linear equation systems of equations.</td><td>2 hours</td></tr>
<tr><td>Numerical differentiation and integration algorithms.</td><td>2 hours</td></tr>
<tr><td>Numerical interpolation.</td><td>2 hours</td></tr>
<tr><td>Optimisation methods.</td><td>2 hours</td></tr>
<tr><td>1st order differential equations.</td><td>2 hours</td></tr>
<tr><td>Final examination.</td><td>2 hours</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">
Bibliography

- I. Rusu, Dana Gavrilescu, Vlad Al. Grosu - "*Programarea calculatoarelor în limbaj C*", Editura MatrixRom, Bucureşti, 2002.
- I. Rusu, Dana Gavrilescu, Vlad Al. Grosu – "*Îndrumar de laborator pentru programarea calculatoarelor: C*", Editura MatrixRom, Bucureşti, 2004.
- I. Rusu, Vlad Al. Grosu – "*Programarea calculatoarelor în limbaj C: probleme rezolvate şi comentate*", Editura MatrixRom, Bucureşti, 2008.
- Brian Kernighan, Dennis Richie – "*The C programming language*", Prentice Hall, New Jersey, ediţiile 1978 şi 1988.
- Herbert Schildt – "*C – manual complet*", Editura Teora, 1999-2003.
- Florin Munteanu, Gh. Muscă, Florin Moraru – "*C - tehnici de programare*", Editura Joint Printing House, Bucureşti, 1995.

</td></tr>
</table>

9. Linking the course content to the expectations of the epistemic community and from the most representative professional associations and employers for the domain of the program

Nowadays the numerical algorithms area requires well-prepared future researchers and developers. Therefore, the students have to have strong backgrounds in the mathematical methods applied in this field with respect to the design, testing and signal processing contexts. The students' preparation on a common programming branch has a well-defined purpose: offer the fundamental backgrounds necessary in any future professional activities.

The *Computers programming* related teaching activities offers the backgrounds of the algorithmic thinking as well as the fundamentals required by any programming language one can use today. The ANSI C programming language is a well-maintained and developed one, its latest standard being ISO/IEC

9899:2011.
According to the curriculum's schedule and the tutoring activities, the offered information follows the necessary steps required to identify and highlight each program's, project's, concept's, method's or theory's qualities and limitations.
Politehnica University of Bucharest is already part of the most recent European Union's academic regulations. The adequate usage of evaluation criteria and methods - in agreement with these regulations - offer the meanings for students' continuous self-evaluation. The evaluation process relates to the marks as well to the methodological notes and directives that the lectures/laboratories holders offer.

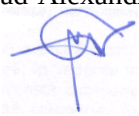
10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Contribution to the final mark (%)
10.4 Lecture	Correct identification of theoretical and practical conditions required by the presented methods and techniques. Diving into specific notions of algebra and mathematical analysis required in Electronics (fundamental preparation domain). C4.1: Definition of the concepts, principles and methods used in: computer programming, high-level programming languages, computational systems architecture, programmable electronic systems, computer graphics and reconfigurable software architectures.	Final examination. Covers both the theoretical and computational aspects presented throughout the semester. The final timed examination requires analysis and synthesis skills in order to find solutions of typical programming problems by applying the suitable algorithms.	50%
10.5 Laboratory	C4.5: Attend and promote an examination test that involves concepts related to the architecture and functional principles of a functional *software structure*. C6.5: Attend an examination test that involves the ability to establish and then describe all the necessary operations involved by the implementation and testing	Constant review of the presented concepts, through short timed tests. The students can accumulate: − 15%, as a result of short quizes (10' per test); − 10%, from homework. − 35%, final examination. The lab ends with a *final examination*, taken individually, each student having a dedicated workstation. One has to: - write an ANSI C program that	50%

	of a typical programming algorithm.	correctly implements the algorithm required in order to solve a certain problem. - give short answer to theoretical question that synthesizes the concepts presented throughout the semester.	
10.6 Minimum performance standard:			
Check the analysis and synthesis skills in identification of practical conditions of the lectured methods as well as in solving typical programming problems. In order to promote the course one has to accumulate at least 50 points out of the total amount of 100 points. No intermediate thresholds are required.			

Date,
25.09.2017

Lecturer and Instructor for practical activities,
Ş.l. dr. ing. Vlad-Alexandru GROSU



Date of Department's approval,
26.09.2017

Director of Department,
Conf. dr. ing. Marian VLĂDESCU