**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 23. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Systems Theory | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | Assoc. prof. dr. eng. Paula Raica – Paula.Raica@aut.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | Sl.dr.ing. Ionut Muntean – Ionut.Muntean@aut.utcluj.ro<br>Sl.dr.ing. Lucian Busoniu – Lucian.Busoniu@aut.utcluj.ro,<br>Sl.dr.ing. Cosmin Marcu – Cosmin.Marcu@aut.utcluj.ro | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | exam | 2.8 | Subject category | DID/OB |

3. Estimated total time

| Sem. | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Systems Theory** | **2** | **-** | **2** | **-** | **28** | **-** | **28** | **-** | **48** | **104** | **4** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 4 | 3.2 | of which, course | 2 | 3.3 | applications | 2 |
| 3.4 | Total hours in the teaching plan | 56 | 3.5 | of which, course | 28 | 3.6 | applications | 28 |

| Individual study | Hours |
|---|---|
| Manual, lecture material and notes, bibliography | 20 |
| Supplementary study in the library, online and in the field | 5 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | 20 |
| Tutoring | |
| Exams and tests | 3 |
| Other activities | |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 48 |
| 3.8 | Total hours per semester | 104 |
| 3.9 | Number of credit points | 4 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | Mathematical Analysis_II (Integral calculus and differential equations, Linear algebra |
| 4.2 | Competence | Differential equations, complex numbers, Laplace transform, linear algebra |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | N/A |
| 5.2 | For the applications | Reading and understanding of the lecture notes. |

6. Specific competences

| | |
|---|---|
| Professional competences | **C1 –** Operating with basic Mathematical, Engineering and Computer Science concepts (4 credits)<br>**C1.1 –** Recognizing and describing concepts that are specific to the fields of calculability, complexity, programming paradigms, and modeling computational and communication systems<br>**C1.2 –** Using specific theories and tools (algorithms, schemes, models, protocols, etc.) for explaining the structure and the functioning of hardware, software and communication systems<br>**C1.4 –** Formal evaluation of the functional and non-functional characteristics of computing systems<br>**C1.5 –** Providing a theoretical background for the characteristics of the designed systems |
| Cross competences | N/A |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | The general objective of the course is to introduce the fundamental principles of linear system modeling, analysis and feedback control and to evaluate feedback control systems with desired behavior. |
| 7.2 | Specific objectives | The specific objectives are to acquire the knowledge and techniques related to:<br>- mathematical system modeling (differential equations, input-output representation as transfer functions, block diagrams) for simple applications<br>- linear system analysis (assessment of stability and performance properties of linear systems) in time and frequency domains<br>- design of feedback controllers such as PID, lead and lag compensators for linear systems using s-domain techniques<br>- linear sampled-data system representation and analysis |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Introduction to system theory and control engineering | Lecture, visual presentations, demonstrations | N/A |
| 2 | Mathematical models of systems. Transfer functions and system response | | |
| 3 | Block diagram models. Block diagram reduction. MIMO systems | | |
| 4 | Analysis of linear continuous systems. 1st and 2nd order systems. Steady-state error. | | |
| 5 | Higher order systems. Stability of linear continuous systems | | |
| 6 | System analysis using root locus | | |
| 7 | Frequency response. Bode diagrams. | | |
| 8 | Frequency response. Stability in the frequency domain. Applications | | |
| 9 | PID – the basic technique for feedback control; | | |
| 10 | Controller design using root locus. Lead-lag compensators | | |
| 11 | Controller design using root locus. Applications | | |
| 12 | Sampled-data systems | | |
| 13 | Digital control systems | | |
| 14 | Sampled data and digital control systems. Applications | | |

Bibliography

1. R. C. Dorf, R. Bishop, “Modern Control Systems”, Addison-Wesley, 2004;
2. K. Ogata , “Modern Control Engineering”, Prentice Hall, 1990.
3. K. Dutton, S. Thompson, B. Barraclough, “The Art of Control Engineering”, Addison-Wesley, 1997
4. William S. Levine (editor), “The Control Handbook”, CRC Press and IEEE Press, 1996
5. Lecture notes available on the course webpage: http://rrg.utcluj.ro/ts

| 8.2. Applications (Seminars, Laboratory, Projects) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Introduction to Matlab. Simulation of dynamical systems | Class discussion, Supervised exercise solving | 4 hours |
| 2 | Linear approximation of differential equations. Transfer functions. System response. | | 4 hours |
| 3 | Block diagram models. 1st and 2nd order system analysis. Steady-state | | 4 hours |

| | | | |
|---|---|---|---|
| | error | using Matlab Miniprojects – individul student reports | |
| 4 | System stability. Root locus | | 4 hours |
| 5 | Frequency response. Bode diagrams | | 4 hours |
| 6 | PID. Lead-lag compensation | | 4 hours |
| 7 | Sampled-data systems | | 4 hours |

Bibliography

1. Paula Raica, “Control Engineering. Exercises”, Editura Mediamira, 2001
2. Lecture notes available on the course webpage: http://rrg.utcluj.ro/ts

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

The course content combines theoretical knowledge with applications and focuses on the formulation and solution of specific problems that may occur in various engineering fields. Application of the control theory concepts are specific to most of the engineering disciplines. The course level is introductory and the intent is to motivate and prepare students for further study in related areas and to conduct projects in real-life applications.

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | Ability to solve exercises related to linear system modeling and analysis | | Midterm exam | | 40% |
| | | Ability to solve exercises related to system design and analysis of sampled-data systems | | Final exam | | 60% |
| Applications | | Answer simple questions from the topic of the lab applications | | Lab tests (optional) | | 20% |
| | | Submitting and defending a miniproject on a given subject | | Individual student report (optional) | | 20% |
| 10.4 Minimum standard of performance | | | | | | |
| Solution of simple exercises applying the knowledge and techniques presented in the course | | | | | | |

Course responsible
Conf.dr.ing. Paula Raica

Head of department
Prof.dr.ing. Rodica Potolea

**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 24. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Computer Architecture | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | S.l.dr.eng. Mihai Negru – Mihai.Negru@cs.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | Conf.dr. eng. Florin Oniga, S.l.dr.eng. Mihai Negru, { Florin.Oniga, Mihai.Negru }@cs.utcluj.ro | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | exam | 2.8 | Subject category | DID/OB |

3. Estimated total time

| Sem. | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Computer Architecture** | **2** | **-** | **2** | **-** | **28** | **-** | **28** | **-** | **74** | **130** | **5** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 4 | 3.2 | of which, course | 2 | 3.3 | applications | 2 |
| 3.4 | Total hours in the teaching plan | 56 | 3.5 | of which, course | 28 | 3.6 | applications | 28 |

| Individual study | Hours |
|---|---|
| Manual, lecture material and notes, bibliography | 28 |
| Supplementary study in the library, online and in the field | 14 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | 28 |
| Tutoring | 0 |
| Exams and tests | 4 |
| Other activities | 0 |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 74 |
| 3.8 | Total hours per semester | 130 |
| 3.9 | Number of credit points | 5 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | 1. Logic design<br>2. Digital system design |
| 4.2 | Competence | Ability to design digital circuits and to implement them in VHDL |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | blackboard, video projector, laptop |
| 5.2 | For the applications | desktop/laptop computer, Xilinx ISE / VIVADO, FPGA development boards |

6. Specific competences

| | |
|---|---|
| Professional competences | **C2 –** Designing hardware, software and communication components (5 credits)<br>**C2.1 –** Describing the structure and functioning of computational, communication and software components and systems<br>**C2.2 –** Explaining the role, interaction and functioning of hardware, software and communication components<br>**C2.3 –** Building the hardware and software components of some computing systems using algorithms, design methods, protocols, languages, data structures, and technologies<br>**C2.4 –** Evaluating the functional and non-functional characteristics of the computing systems using specific metrics<br>**C2.5 –** Implementing hardware, software and communication systems |
| Cross competences | N/A |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | Knowing and understanding the concepts of organization and functioning for central processing units, memories, input/output, and using these concepts for design. |
| 7.2 | Specific objectives | • Applying methods for representation and design at system level for digital circuits<br>• Instruction Set Architecture (ISA) specification<br>• Writing simple programs in assembly languages and machine code<br>• Specification, design, implementation, and testing of Central Processing Units (CPU) – micro architecture – data path – command units<br>• Understanding memory organization and I/O operations<br>• Understanding modern trends in computer architectures |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Introduction | Oral presentation backed up by multimedia equipment, interactive communication, blackboard problem solving | |
| 2 | High-Level Synthesis | | |
| 3 | Instruction Set Architecture (ISA) | | |
| 4 | CPU Design - Single Cycle CPU | | |
| 5 | Computer Arithmetic and Simple Arithmetic Logic Units | | |
| 6 | CPU Design - Multi Cycle CPU Data path | | |
| 7 | CPU Design - Multi Cycle CPU Control | | |
| 8 | CPU Design – Pipelined CPU | | |
| 9 | Advanced Pipelining – Static and Dynamic Scheduling of the Execution | | |
| 10 | Branch Prediction | | |
| 11 | Superscalar Architectures | | |
| 12 | Memory | | |
| 13 | I/O and Interconnection Structures | | |
| 14 | Problem solving | | |

Bibliography


1. D. A. Patterson, J. L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”,5th edition, ed. Morgan–Kaufmann, 2013.
2. D. A. Patterson and J. L. Hennessy, “Computer Organization and Design: A Quantitative Approach”,5th edition, ed. Morgan-Kaufmann, 2011.
3. Vincent P. Heuring, et al., “Computer Systems Design and Architecture”, Addison-Wesley, USA, 1997.
4. A. Tanenbaum, “Structured Computer Organization”, Prentice Hall, USA, 1999.
5. MIPS32 Architecture for Programmers, Volume I: “Introduction to the MIPS 32™ Architecture”.
6. MIPS32 Architecture for Programmers, Volume II: “The MIPS 32™ Instruction Set”.


**Online bibliography**

| |
|---|
| M. Negru, F. Oniga, S. Nedevschi, Lecture slides http://users.utcluj.ro/~negrum |

| 8.2. Applications (Laboratory) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Introduction in the Xilinx ISE environment and the FPGA development board | Blackboard quick overview of key issues, exercises, experimenting with FPGA development boards with specialized IDEs for circuit design and implementation (Xilinx ISE) | |
| 2 | Design and Implementation of Combinational CPU Components | | |
| 3 | Design and Implementation of Sequential CPU Components | | |
| 4 | Design of a Single Cycle CPU 1 (MIPS) | | |
| 5 | Design of a Single Cycle CPU 2 (MIPS) | | |
| 6 | Design of a Single Cycle CPU 3 (MIPS) | | |
| 7 | Design of a Single Cycle CPU 4 (MIPS) | | |
| 8 | Midterm practical evaluation on the FPGA board | | |
| 9 | Pipelined CPU Design | | |
| 10 | Pipelined CPU Design | | |
| 11 | Pipelined CPU Design | | |
| 12 | Pipelined CPU interfacing | | |
| 13 | Practical evaluation of the pipelined CPU on the FPGA board | | |
| 14 | Final Tests and Evaluation | | |
| Bibliography<br>***Online bibliography***<br>1. M. Negru, F. Oniga, S. Nedevschi, Laboratory guide http://users.utcluj.ro/~negrum | | | |

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

| |
|---|
| Computer Architecture is one of the fundamental subjects of the Computer Science and Information Technology field. It combines fundamental and practical aspects used for digital circuits design and implementation. The content of this subject is harmonized with the specific curricula of other national and international universities, and is evaluated by the Romanian government agencies (CNEAA and ARACIS). The practical aspects involve getting familiar with and using development products and tools provided by companies from Romania, Europe, and USA (ex. Xilinx, Digilent). |

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | Testing the theoretical knowledge, the ability of problem solving, presence and activity | | Written exam | | 50 % |
| Applications | | Practical ability to solve and implement specific problems related to processor design, presence and activity | | Lab exam, periodical assessment of results | | 50 % |
| 10.4 Minimum standard of performance | | | | | | |
| Knowing the fundamental theory of the subject, the ability to design and implement a processor with a reduced set of instructions | | | | | | |

Course responsible
S.l.dr. ing. Mihai Negru

Head of department
Prof.dr.eng. Rodica Potolea

**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 25. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Fundamental Algorithms | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | Prof. dr. eng. Rodica Potolea – Rodica.Potolea@cs.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | Assoc. prof. dr. eng. Tudor Mureşan – Tudor.Muresan@cs.utcluj.ro<br>Sl. dr. eng. Camelia Lemnaru – Camelia.Lemnaru@cs.utcluj.ro | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | exam | 2.8 | Subject category | DID/OB |

3. Estimated total time

| Sem. | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Fundamental Algorithms** | **2** | **1** | **2** | **-** | **28** | **14** | **28** | **-** | **60** | **130** | **5** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 5 | 3.2 | of which, course | 2 | 3.3 | applications | 3 |
| 3.4 | Total hours in the teaching plan | 70 | 3.5 | of which, course | 28 | 3.6 | applications | 42 |

| Individual study | Hours |
|---|---|
| Manual, lecture material and notes, bibliography | 21 |
| Supplementary study in the library, online and in the field | 16 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | 16 |
| Tutoring | 8 |
| Exams and tests | 9 |
| Other activities | |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 60 |
| 3.8 | Total hours per semester | 130 |
| 3.9 | Number of credit points | 5 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | Data Structures, Programming |
| 4.2 | Competence | Learning to develop and test programs |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | Whiteboard, projector, computer |
| 5.2 | For the applications | Networks, C ++ |

6. Specific competences

| | |
|---|---|
| Professional competences | **C3.** Problems solving using specific Computer Science and Computer Engineering tools (5 credit points)<br>**C3.1** Identifying classes of problems and solving methods that are specific to computing systems<br>**C3.2** Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results<br>**C3.3** Applying solution patterns using specific engineering tools and mehods<br>**C3.4** Evaluating, comparatively and experimentally, the available alternative solutions for performance optimization<br>**C3.5** Developing and implementing informatic solutions for concrete problems |
| Cross competences | N/A |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | Acquiring modern study of algorithms: design and analysis |
| 7.2 | Specific objectives | • Learning with the efficient resolution<br>• Learning the methods for evaluating the effectiveness<br>• Acquiring main polynomial algorithms<br>• Acquiring calculation complexity<br>• Specify algorithms to shift the focus of control structures<br>• Learning facile implementation of strict compliance with the pseudocode<br>• Effective implementation of key polynomial algorithms<br>• Practical assessment algorithms efficiency: space and processing time |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Mathematical Foundations: Asymptotical notation, Recurrence | Multimedia tools | |
| 2 | Complexity Classes | | |
| 3 | Sorting and Order Statistics | | |
| 4 | Sorting and Order Statistics (continued) | | |
| 5 | Advanced Data Structures : Hash Tables, Trees | | |
| 6 | Advanced Data Structures: Heaps, Disjoint Sets | | |
| 7 | Design and Analysis Advanced Techniques: Dynamic Programming | | |
| 8 | Design and Analysis Advanced Techniques: Greedy Algorithms | | |
| 9 | Design and Analysis Advanced Techniques: Damping Analyze | | |
| 10 | Graphs: Search in a Graph, Minimal Spanning Tree | | |
| 11 | Graphs: Shortest path | | |
| 12 | Graphs: Max Flow | | |
| 13 | Graphs: Bipartite Graphs | | |
| 14 | Learn the basic Complexity sets and representative problems | | |

Bibliography

1. T. Cormen, C. Rleiserson, R. Rivest, C. Stein, *Introduction to Algorithms, Second Edition*, The MIT Press, 2001

| 8.2. Applications (Seminars, Laboratory) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Efficient implementation and comparison of sorting algorithms | Assistance and verification practice | |
| 2 | Efficient implementation and comparison of sorting algorithms (continued) | | |
| 3 | Efficient implementation and comparison of lists algorithms | | |
| 4 | Efficient implementation and comparison of lists algorithms (continued) | | |
| 5 | Efficient implementation and comparison of trees algorithms | | |
| 6 | Efficient implementation and comparison of trees algorithms (continued) | | |
| 7 | Implementation of augmented data structures | | |
| 8 | Implementation of augmented data structures (continued) | | |
| 9 | Efficient implementation of graphs algorithms | | |
| 10 | Efficient implementation of graphs algorithms (continued) | | |
| 11 | Efficient implementation of graphs algorithms (continued) | | |

| | | | |
|---|---|---|---|
| 12 | Efficient implementation of graphs algorithms (continued) | | |
| 13 | Approximation algorithms | | |
| 14 | Final Evaluation | | |

Bibliography

1. T. Cormen, C. Rleiserson, R. Rivest, C. Stein, *Introduction to Algorithms, Second Edition*, The MIT Press, 2001

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

Discipline is fundamental in the field of Computer and Information Technology, its content is beyond dispute, familiarizing students with the principles of design and analysis of algorithms. Course content is aligned all departments of computer science in the world and was rated by Romanian governmental agencies (CNEAA and ARACIS).

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | Theoretical problem solving skills | | Written exam | | 70% |
| Applications | | Problem solving skills practice | | Written exam | | 30% |
| 10.4 Minimum standard of performance | | | | | | |
| final grade ≥5 | | | | | | |

Course responsible
Prof.dr.eng. Rodica Potolea

Head of department
Prof.dr.eng. Rodica Potolea

**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 26. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Fundamental Programming Techniques | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | Prof. dr. eng. Ioan Salomie - Ioan.Salomie@cs.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | Conf. dr.eng. Viorica Chifu – Viorica.Chifu@cs.utcluj.ro<br>S.l. dr. eng. Tudor Cioară – Tudor.Cioara@cs.utcluj.ro<br>S.l. dr. eng. Ionut Anghel – Ionut.Anghel@cs.utcluj.ro<br>S.l. dr. eng. Cristina.Pop – Cristina.Pop@cs.utcluj.ro | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | exam | 2.8 | Subject category | DF/OB |

3. Estimated total time

| Sem. | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Fundamental Programming Techniques** | **2** | **-** | **2** | **-** | **28** | **-** | **28** | **-** | **74** | **130** | **5** |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 4 | 3.2 | of which, course | 28 | 3.3 | applications | 28 |
| 3.4 | Total hours in the teaching plan | 56 | 3.5 | of which, course | 28 | 3.6 | applications | 28 |

| Individual study | Hours |
|---|---|
| Manual, lecture material and notes, bibliography | 18 |
| Supplementary study in the library, online and in the field | 16 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | 24 |
| Tutoring | |
| Exams and tests | 16 |
| Other activities | |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 74 |
| 3.8 | Total hours per semester | 130 |
| 3.9 | Number of credit points | 5 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | Object Oriented Programming Fundamentals |
| 4.2 | Competence | Knowledge of Object Oriented Programming |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | Whiteboard, projector, computer |
| 5.2 | For the applications | Computers, specific software |

6. Specific competences

| | |
|---|---|
| Professional competences | **C4 -** Improving the performances of the hardware, software and communication systems<br>**C4.1** - Identifying and describing the defining elements of the performances of the hardware, software and communication systems<br>**C4.2** - Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems<br>**C4.3 -** Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems<br>**C4.**4 - Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems<br>**C4.5** - Developing professional solutions for hardware, software and communication systems based on performance optimization |
| Cross competences | N/A |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | Knowledge of and application of object-oriented programming techniques in developing applications sofware |
| 7.2 | Specific objectives | - Be able to use programming techniques for designing classes and interfaces, including contracts and invariant<br>- Be able to use programming techniques for code reuse using inheritance and polymorphism<br>- Be able to use generic programming techniques for processing collections<br>- Be able to use programming techniques and event-based reflection<br>- Be able to use programming techniques and concurrent multi-threading<br>- Be able to use design patterns and frameworks for design reuse solutions<br>- Be able to use programming techniques for performance and maintenance software |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Class level design and implementation techniques | Using modern multimedia teaching methods and direct access to internet<br>Students are invited to collaborate in research projects<br>Tutorials and personal assistance hours the semester and before the exam | |
| 2 | Programming techniques with classes and interfaces | | |
| 3 | Programming techniques using inheritance and polymorphism | | |
| 4 | Programming techniques using contracts and invariants | | |
| 5 | Generic programming techniques | | |
| 6 | Reflection techniques | | |
| 7 | Event-driven techniques | | |
| 8 | Collection programming techniques | | |
| 9 | Concurrent and multithreading techniques | | |
| 10 | Flexibility and reuse through design patterns | | |
| 11 | Main design patterns of type creational, structural and behavioral | | |
| 12 | Flexibility and reuse through frameworks | | |
| 13 | Techniques for efficiency and performance | | |
| 14 | Techniques for clarity and maintenance | | |

Bibliography

1. Ioan Salomie - Tehnici Orientate Obiect, Editura Albastra, Microinformatica, 1995
2. Eric Gamma, Helm, Johnson, Vlissides - Design Patterns, Addison Wesley, 1995 (translated into Romanian by Teora Publ. as "Sabloane de Proiectare")
3. Joshua Bloch - Effective Java, 2/e Addison Wesley, 2008
4. Xiaoping Jia - Object Oriented Software Development using Java, Addison Wesley, 2002
5. Ioan Salomie, Note de Curs, http://www.coned.utcluj.ro/~salomie/TP

| 8.2. Applications (Seminars, Laboratory, Projects) | Teaching methods | Notes |
|---|---|---|

| | | | |
|---|---|---|---|
| 1-2 | Programming with classes and objects using Java, C# and C++ (2 lab sessions) | Short presentation of the themes laboratory, discussions on themes, themes implementing computer, personal computer miniproject | |
| 3-4 | Programming with inheritance and polymorphism using Java, C# and C++ (2 lab sessions) | | |
| 5-6 | Programming with Java Collection Framework (2 lab sessions) | | |
| 7-8 | Error handling in Java programming (2 lab sessions) | | |
| 9-10 | Multi-threading programming in Java (2 lab sessions) | | |
| 11-12 | Mini-project (involving design patterns and frameworks) (2 lab sessions) | | |
| 13-14 | Laboratory Test and mini-project presentations | | |

Bibliography

1. Steve McConnell - Code Complete, 2/e, Microsoft Press, 2004
2. http://java.sun.com/docs/books/tutorial

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

It's a discipline domain "Computer and Information Technology". She instructs students in the application of object-oriented programming techniques in designing and implementing applications sofware. Course content was determined by analyzing equivalent disciplines from other universities and based on the requirements of IT employers in Romania. Course content also was assessed by Romanian governmental agencies (CNEAA and ARACIS).

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | -Ability to use of object-oriented programming techniques in designing and implementing applications sofware<br>-Ability to use of design patterns and frameworks sites for reuse design solutions<br>- Ability to solve problems using object oriented programming techniques | | written exam | | 55% |
| Applications | | - Problem solving skills using object oriented programming techniques<br>- Presence Activity | | written exam | | 45% |

10.4 Minimum standard of performance

- Can use object oriented programming techniques in designing and implementing software applications
- Obtaining final grade 5.

Course responsible
Prof.dr.eng. Ioan Salomie

Head of department
Prof.dr.eng. Rodica Potolea

**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 27. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Operating Systems | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | Lect. dr. eng. Adrian Coleşa – adrian.colesa@cs.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | Lect. dr. eng. Adrian Coleşa – adrian.colesa@cs.utcluj.ro<br>Eng. Gheorghe Hajmasan – ghajmasan@bitdefender.com<br>Eng. Andrei Lutas – vlutas@bitdefender.com<br>Eng. Radu Ciocas – rciocas@bitdefender.com | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | exam | 2.8 | Subject category | DID/OB |

3. Estimated total time

| Sem. | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Operating Systems** | **2** | **-** | **2** | **-** | **28** | **-** | **28** | **-** | **74** | **130** | **5** |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 4 | 3.2 | of which, course | 2 | 3.3 | applications | 2 |
| 3.4 | Total hours in the teaching plan | 56 | 3.5 | of which, course | 28 | 3.6 | applications | 28 |
| Individual study | | | | | | | | Hours |
| Manual, lecture material and notes, bibliography | | | | | | | | 30 |
| Supplementary study in the library, online and in the field | | | | | | | | 10 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | | | | | | | | 28 |
| Tutoring | | | | | | | | 2 |
| Exams and tests | | | | | | | | 4 |
| Other activities | | | | | | | | 0 |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 74 |
| 3.8 | Total hours per semester | 130 |
| 3.9 | Number of credit points | 5 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | Computer Programming, Data Structures and Algorithms |
| 4.2 | Competence | C programming |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | Students must have minimum 9 classes attended to be allowed to take the exam |
| 5.2 | For the applications | Students must have minimum 11 classes attended to be allowed to take the exam |

6. Specific competences

| | |
|---|---|
| Professional competences | **C3:** Problems solving using specific Computer Science and Computer Engineering tools (3 credits)<br>• **C3.1**: Identifying classes of problems and solving methods that are specific to computing systems<br>• **C3.2**: Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results<br>• **C3.3**: Applying solution patterns using specific engineering tools and methods<br>• **C3.4**: Evaluating, comparatively and experimentally, the available alternative solutions for performance optimization<br>• **C3.5**: Developing and implementing informatic solutions for concrete problems<br>**C4**: Improving the performances of the hardware, software and communication systems (2 credits)<br>• **C4.1**: Identifying and describing the defining elements of the performances of the hardware, software and communication systems<br>• **C4.2**: Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems<br>• **C4.3**: Applying the fundamental methods and principles for increasing the performances of the hardware, software and communication systems<br>• **C4.4**: Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems<br>• **C4.5**: Developing professional solutions for hardware, software and communication systems based on performance optimization |
| Cross competences | N/A |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | Have a clear understanding of what an OS is, its role and general functionality and be able to use most of the OS system calls. |
| 7.2 | Specific objectives | Have knowledge and understand the general structure and functionality of an OS.<br>Understand the specific functionality of the most important OS components, like shell, process manager, file system, memory manager, security manager.<br>Understand the functionality of main synchronization mechanisms and be able to use them to solve real synchronization problems.<br>Be able to write C programs to use an OS (Linux and Windows) system calls. |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | **Introduction and basic concepts**. OS's definition, role, evolution, components, main concepts (file, process, system calls). Basic hardware aspects: CPU, user and kernel mode, memory layers, I/O devices. Basic OS structure. | (1) lecture presentation based on beamer presentation;<br>(2) interactions with students: ask their opinion relative to the presented subject;<br>(3) give each class a short evaluation test; let students discuss and argue each other their solution; give them the good solution and let them evaluate their own one;<br>(4) propose 2-3 interesting study cases of OSes to be prepared and presented by students;<br>(5) students are | |
| 2 | **The Shell (Command Interpreter)**. Definition, role, functionality, simple and complex commands. Standard input and output redirection. | | |
| 3 | **File systems (1)**. User Perspective. File and directory concept from the user point of view (definition, role, characteristics, operations). | | |
| 4 | **File systems (2)**. Windows and Linux File Systems. Permission rights and system calls. | | |
| 5 | **File systems (3)**. Implementation aspects. Implementation strategies overview, space management and related problems, hard and symbolic links. | | |
| 6 | **Process management**. Process model: definition, role, characteristics. Linux and Windows process management system calls. | | |
| 7 | **Thread management**. Thread model: user vs. kernel threads, implementation problems, usage, performance aspects. Basic scheduling algorithms (FIFO, SJF, Priority-based). Linux and Windows process thread system calls. | | |
| 8 | **Process synchronization (1)**. Theoretical aspects. Context, definition, synchronization mechanisms, techniques and problems (locks, semaphores, monitors, mutual exclusion, starvation, deadlock). | | |
| 9 | **Process synchronization (2)**. Classical synchronization patterns: producer/consumer, readers/writers, rendez-vous, barrier, dining | | |

| | | | |
|---|---|---|---|
| | philosopher, sleeping barber. Similarities between different synchronization mechanisms. | invited to collaborate in research projects. | |
| 10 | **Inter-process communication**. Pipe files, shared memory, message queues, signals. | | |
| 11 | **Memory management (1)**. Context, definition, binding, basic techniques, space management, addresses translation, swapping. | | |
| 12 | **Memory management (2)**. Paging and segmentation. | | |
| 13 | **I/O Devices Management**. Principles, disks, clocks, character-oriented terminals. | | |
| 14 | **Security aspects**. Security policies and mechanisms. Basic program's vulnerabilities (buffer overflow). | | |

Bibliography

1. Andrew Tanenbaum. *Modern Operating System*, 2nd Edition, Prentice-Hall, 2005, ISBN 0-13-092641-8.

| 8.2. Applications (Laboratory) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Linux File System | (1) students are presented a very brief overview of the most important and difficult aspects of the working subject ;<br>(2) students are given at the beginning of each class a short evaluation quiz;<br>(3) students are given a hands-on tutorial to practice with working subject's aspects and to solve problems<br>(4) students are given challenging problems for extra credit; | |
| 2 | Linux Commands | | |
| 3 | Linux Shell Scripts | | |
| 4 | Linux System Calls for File Access | | |
| 5 | Linux System Calls for File and Directory Manipulation | | |
| 6 | Windows File System (NTFS) | | |
| 7 | Linux Processes | | |
| 8 | Linux Threads | | |
| 9 | Windows Processes and Threads | | |
| 10 | Linux Semaphores | | |
| 11 | Linux Locks and Condition Variables | | |
| 12 | Linux pipes | | |
| 13 | Memory management. | | |
| 14 | Security aspects. Buffer overflow detection and correction. | | |

Bibliography

Lecture slides and laboratory text and support at http://os.obs.utcluj.ro/moodle

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

OS knowledge is a fundamental requirement in the CS field. We follow the CAM curricula guide. We also consult IT companies about their practical expecations regarding OS knowledge and adapt accordingly our course contents. In this sense, Linux and Windows are the most used Oses.

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | Check if students: understand fundamental OS concepts and are able to recognize an OS-related problem and find solution to it. | | Small problem-like subject requiring students to apply the theoretical OS aspects to give a solution to that problem. | | 0.67 |
| Applications | | Check if students are able to write C programs to use different OS provided system calls to solve practical problems. | | Quiz tests. Programming problems, whose solution has to be implemented in C and run on computers. | | 0.33 |

10.4 Minimum standard of performance.

Know the basic system calls, understand their functionality and be able to use them.

Course responsible
Lect. dr. eng. Adrian Coleşa

Head of department
Prof.dr.eng. Rodica Potolea

**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 28. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Elements of Computer Assisted Graphics | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | Prof. dr. eng. Gorgan Dorian – dorian.gorgan@cs.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | S.l. dr. eng. Melenti Cornelia, S.l. dr. eng. Bacu Victor, {cornelia.melenti, victor.bacu}@cs.utcluj.ro | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | exam | 2.8 | Subject category | DF/OB |

3. Estimated total time

| Sem. | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Elements of Computer Assisted Graphics** | **2** | **-** | **2** | **-** | **28** | **-** | **28** | **-** | **48** | **104** | **4** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 4 | 3.2 | of which, course | 2 | 3.3 | applications | 2 |
| 3.4 | Total hours in the teaching plan | 56 | 3.5 | of which, course | 28 | 3.6 | applications | 28 |

| Individual study | Hours |
|---|---|
| Manual, lecture material and notes, bibliography | 20 |
| Supplementary study in the library, online and in the field | 6 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | 10 |
| Tutoring | 3 |
| Exams and tests | 9 |
| Other activities | 0 |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 48 |
| 3.8 | Total hours per semester | 104 |
| 3.9 | Number of credit points | 4 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | Computer programming (C language) |
| 4.2 | Competence | Applications development in C programming language |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | Projector, computer |
| 5.2 | For the applications | Laboratory attendance is mandatory<br>Study of laboratory materials from the server |

6. Specific competences

| | |
|---|---|
| Professional competences | **C3 –** Problems solving using specific Computer Science and Computer Engineering tools (4 credits)<br>**C3.1 –** Identifying classes of problems and solving methods that are specific to computing systems<br>**C3.2 –** Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results<br>**C3.3 –** Applying solution patterns using specific engineering tools and mehods<br>**C3.4 –** Evaluating, comparatively and experimentally, the available alternative solutions for performance optimization<br>**C3.5 –** Developing and implementing informatic solutions for concrete problems |
| Cross competences | N/A |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | Learning about the architecture of a graphic system, the study of the graphic pipeline, the study of 2D graphic algorithms |
| 7.2 | Specific objectives | 1. Creation of the graphical model of a scene of objects<br>2. Implementation of the basic algorithms that form the core of a graphic system<br>3. Development of graphic applications in a high-level programming language (C, C++)<br>4. Implementation of the main phases of the graphic transformation pipeline |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Introduction. History. Examples | New multimedia teaching approaches will be used in classes.<br><br>The course is interactive and includes demonstrations that exemplify graphical methods and algorithms. | During the semester and before each exam there are a few preparation hours planned. |
| 2 | Graphics systems – architecture, standards | | |
| 3 | Graphics devices – logic and physics devices, input, output and interactive devices | | |
| 4 | Graphics transformations pipeline – 2D and 3D transformations. Matrix operators | | |
| 5 | Mathematics in computer graphics | | |
| 6 | Lines scan conversion algorithms | | |
| 7 | Circles scan conversion algorithms | | |
| 8 | Polygons scan conversion algorithms | | |
| 9 | Clipping algorithms – point, line, polygon and text | | |
| 10 | Projections and viewing transformations | | |
| 11 | Photorealistic presentation of 3D objects – concepts, algorithms, examples | | |
| 12 | Color models – color perception, color space and standards, color in software design | | |
| 13 | Graphics formats – vector and raster formats, data compression , Web technologies | | |
| 14 | Graphics pattern grammars | | |

Bibliography


7. Foley J.D., van Dam, A., Feiner, S.K., Hughes, J.F., "*Computer Graphics. Principles and Practice*". Addison-Wesley Pblishing Comp., 1992.
8. Watt A., "*3D Computer Graphics*". Addison-Wesley, 1998.


***In virtual libarry***

1. Course resurses, http://cgis.utcluj.ro/didactic

| 8.2. Applications (Laboratory) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Basic graphics application structure | Documentation and examples will be available to the students, prior to the laboratory classes, on a dedicated server. | Each student will have to develop a specific project based on |
| 2 | Output and input operations in graphics window | | |
| 3 | Inputs by keyboard, mouse and timer | | |
| 4 | Menu, icon, cursor and bitmap resources – Part 1 | | |
| 5 | Menu, icon, cursor and bitmap resources – Part 2 | | |
| 6 | Coordinate systems. Viewing transformations | | |
| 7 | 2D clipping. Cohen-Sutherland Algorithm | | |

| | | | |
|---|---|---|---|
| 8 | Projections. 2D and 3D transformations | The students will work independently but will also be assisted by the teacher. | the knowledge acquired at the laboratory hours. |
| 9 | Lines scan conversion. Bresenham method | | |
| 10 | Polygon clipping. Sutherland-Hodgman Algorithm | | |
| 11 | Polygon clipping. Weiler Clipping Algorithm | | |
| 12 | Photorealistic presentations | | |
| 13 | Color computation | | |
| 14 | Assessment | | |

Bibliography

***In virtual libarry***

1. Course and practical works, http://cgis.utcluj.ro

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

This discipline is integrated into the Computers and Information Technology domain. The content is classic, yet modern, and introduces to students the fundamentals of graphic systems and 2D algorithms. The content of this discipline has been aligned with the information presented in similar disciplines from other major universities and companies from Romania, Europe and USA and has been evaluated by the authorized Romanian governmental agencies (CNEAA and ARACIS).

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | The written exam tests the understanding of the information presented in classes and the ability to apply this knowledge. The activity in class evaluates the active involvement of the students in the teaching process and their participation to the discussions, debates and other class activities during the entire semester. | | Evaluation is performed through written exam (E) and classes activity (AC) | | 50% (E)<br>10% (AC) |
| Applications | | Laboratory assessment evaluates the practical abilities obtained by the students. Through homework assignments the students have the opportunity to develop their skill in applying the notions, concepts and methods presented in class. | | Evaluation is performed through written exam. | | 40% |
| 10.4 Minimum standard of performance | | | | | | |
| Final mark: N=0,5*E+0,4*[(C+T)/2]+0.1*AC<br>Graduation requirement: N≥5; | | | | | | |

Course responsible
Prof. dr. eng. Dorian Gorgan

Head of department
Prof.dr.eng. Rodica Potolea

**SYLLABUS**

1. Data about the program of study

| | | |
|---|---|---|
| 1.1 | Institution | The Technical University of Cluj-Napoca |
| 1.2 | Faculty | Automation and Computer Science |
| 1.3 | Department | Computer Science |
| 1.4 | Field of study | Computer Science and Information Technology |
| 1.5 | Cycle of study | Bachelor of Science |
| 1.6 | Program of study/Qualification | Computer Science/ Engineer |
| 1.7 | Form of education | Full time |
| 1.8 | Subject code | 29. |

2. Data about the subject

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2.1 | Subject name | | | | | Foreign Language II (English, French, German - Technical documents elaboration) | | | | | |
| 2.2 | Subject area | | | | | Computer Science and Information Technology | | | | | |
| 2.3 | Course responsible/lecturer | | | | | Lect. dr. Sonia Munteanu Sonia.Munteanu@lang.utcluj.ro | | | | | |
| 2.4 | Teachers in charge of applications | | | | | - | | | | | |
| 2.5 | Year of study | II | 2.6 | Semester | 4 | 2.7 | Assessment | Colloquium | 2.8 | Subject category | DC/OB |

3. Estimated total time

| Sem | Subject name | Lecture | Applications | | | Lecture | Applications | | | Individual study | TOTAL | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [hours / week.] | | | | [hours / semester] | | | | | | |
| | | | S | L | P | | S | L | P | | | |
| **4** | **Foreign Language II (English, French, German - Technical documents elaboration)** | **2** | **-** | **-** | **-** | **28** | **-** | **-** | **-** | **24** | **52** | **2** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3.1 | Number of hours per week | 2 | 3.2 | of which, course | 2 | 3.3 | applications | - |
| 3.4 | Total hours in the teaching plan | 28 | 3.5 | of which, course | 28 | 3.6 | applications | - |

| Individual study | Hours |
|---|---|
| Manual, lecture material and notes, bibliography | 6 |
| Supplementary study in the library, online and in the field | 8 |
| Preparation for seminars/laboratory works, homework, reports, portfolios, essays | 6 |
| Tutoring | 0 |
| Exams and tests | 4 |
| Other activities | |

| | | |
|---|---|---|
| 3.7 | Total hours of individual study | 24 |
| 3.8 | Total hours per semester | 52 |
| 3.9 | Number of credit points | 2 |

4. Pre-requisites (where appropriate)

| | | |
|---|---|---|
| 4.1 | Curriculum | B1 according to the Common European Framework for Languages |
| 4.2 | Competence | Continuous training |

5. Requirements (where appropriate)

| | | |
|---|---|---|
| 5.1 | For the course | Subjects of semester 1, year 2 |
| 5.2 | For the applications | |

6. Specific competences

| | |
|---|---|
| Professional competences | N/A |
| Cross competences | **CT3 –** Demonstrating the spirit of initiative and action for updating professional, economical and organizational culture knowledge (2 credits) |

7. Discipline objectives (as results from the *key competences gained*)

| | | |
|---|---|---|
| 7.1 | General objective | Development of integrated skills in an engineering professional context |
| 7.2 | Specific objectives | - Mastering documenting strategies, information processing; writing according to discourse patterns in specific purposes contexts;<br>- Acquiring strategies for handling difficult written text on a variety of science related topics;<br>- Comprehension and production of discipline appropriate text and genre.<br>- Use of lexical and grammar structures at B1/B2 language competence levels, according to CEFL. |

8. Contents

| 8.1. Lecture (syllabus) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | Hierarchical structure of grammar. Natural language processing; morphology, syntax, discourse. Language knowledge in technology development for language processing and artificial intelligence. | lecture, conversation, elicitation, practical application of knowledge, assignment discussion | |
| 2 | Student's research on NLP and AI topics which involve knowledge about language. Assignment discussion. | | |
| 3 | Word structure: inflected and derivated words. Derivation as a means of creating technical vocabulary. | | |
| 4 | Phrases: noun headed phrases, verb headed phrases, adjective headed phrases, preposition headed phrases. | | |
| 5 | Simple and complex sentences. Frequently used phrase/sentence structures in technical texts: coordination and subordination in finite and non-finite clauses. | | |
| 6 | Cohesion and coherence in discourse. Readibility of technical texts: syntactic parallelism, sentence rephrasal, nominalization, lexical choice | | |
| 7 | Structure of information in paragraphs: general-particular patterns, theme-rheme, hypothesis and validation. | | |
| 8 | Mid term evaluation.. | | |
| 9 | The informative function of science discourse: information structure, impersonal expression, nominalized theme. | | |
| 10 | Functional and rhetorical organization of written science discourse: genres (textbooks, journal articles and scientific posters).. | | |
| 11 | Information organization in scientific posters: functional and structural patterns. Design options and selection of information – from the journal article to the poster. | | |
| 12 | Formulaic language in science discourse: multifunctional lexical bundles. Interpersonal function of science discourse: hedges, boosters and author mention in science discourse. | | |
| 13 | Disciplinary variation in science discourse: professional communities, discourse communities. Selecting from language resources according to disciplinary practices. | | |

| | | | |
|---|---|---|---|
| 14 | Test. | | |

Bibliography

1.Munteanu, S.-C (2013) *Academic English for Science and Engineering*. Cluj-Napoca: Casa Cartii de Stiinta. ISBN 978-606-17-0398-2.
2.Swales John M. & Christine B. Feak (2001) *Academic Writing For Graduate Students - Essential Tasks And Skills*, Ann Arbor*:* The University Of Michigan Press.
3. Hyland Ken (2006) *English For Academic Purposes - An Advanced Resource Book,* London: Routledge

| 8.2. Applications (Seminars, Laboratory, Projects) | | Teaching methods | Notes |
|---|---|---|---|
| 1 | - | | |

Bibliography

-

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

Mastering a foreign language will support students in a more flexible integration in the labour market, and have improved personal development. The introduction in the language for specific purposes will facilitate reading more documents in the field of study.

10. Evaluation

| Activity type | 10.1 | Assessment criteria | 10.2 | Assessment methods | 10.3 | Weight in the final grade |
|---|---|---|---|---|---|---|
| Course | | Assessment and homework completion in due time;<br>Ability to comprehend below and above sentence syntactic and morphologic structures specific to science discourse; to read from sources, to comprehend complex text (journal articles, textbooks); write and present a poster. | | Oral presentation of homework and assignment tasks;<br>Two written tests or one test and science poster. | | assignment/ homework = 20%<br>mid-term test = 30%<br>poster/final test = 40%<br>poster presentation =10%<br>total = 100% |
| Applications | | | | | | |

10.4 Minimum standard of performance

Minimum 60% of the final test, regarding language, lexical and discourse structures used in the technical discourse, linking words, verbs in impersonal moods, nominal groups, revision and correction of written texts.
Assignment completion, minimum 50% of the mid term evaluation.

Course responsible
Lect. dr. Sonia Munteanu

Head of department
Prof.dr.eng. Rodica Potolea