Study Guide
I. GENERAL INFORMATION ABOUT THE FACULTY

1. Name and address
"Alexandru Ioan Cuza" University
Faculty of Computer Science
Str. General Berthelot nr. 16
700483 Iaşi
Romania
http://www.infoiasi.ro

2. Short history and mission

History
The interest for Computer Science at the "Alexandru Ioan Cuza" University of Iaşi dates back in 1958-1965, when Adolf Haimovici, professor at the Faculty of Mathematics, initiated a series of lectures about "Mathematics Applied in Computer Science"; in 1960, professor Haimovici taught the first course entitled "Elements of Computer Science".

In 1965, the Section of Computing Machines was set up within the Faculty of Mathematics; its first students graduated in 1970. In 1971, the section changed its name in the Section of Computer Science.

The year 1975 witnessed the creation of the Computing Centre of the University. During the leadership of professor Călin Petru Ignat, the first head of the Centre, were created the conditions for the development of research activity at the Computing Centre and thus for the emergence, in time, of the Faculty of Computer Science. Continuing this process of construction, professor Toader Jucan - head of the Centre since 1981 - pursued a rigorous policy regarding the presonnel's selection and training; many of today's professors of the faculty owe him their academic careers. The following heads of the Centre, especially Cornelius Croitoru and Gheorghe Grigoraş, continued on the same direction, until the Centre has been integrated in the Faculty of Computer Science.

As a result of the new realities - the significant growth of the number of students in computer science, the necessity for a continuous and quick adaptation of the teaching plans to the permanent changes in the field of computer science -, in the Autumn of 1991 the teaching staff of the Department of Computer Science of the Faculty of Mathematics proposed the creation of a new faculty, based on the structure of the existing section. In December, the Senate of the University approved the creation of the new faculty. The representatives to the Senate of the future Faculty of Computer Science were, by that time, professor Călin Ignat (Rector of the University) and associate professor Cornelius Croitoru. Thorough the decision of the Ministry of Education and Science, in January 1992 was set up the Faculty of Computer Science of the "Alexandru Ioan Cuza" University of Iaşi.


The initial structure of the faculty included two departments: the Department of Theoretical Computer Science and the Department of Applied Computer Science. Between 2004 and 2007 there were three departments: the Department of Computer Fundamentals and Distributed Systems, the Department of Optimization and Artificial Intelligence, and the Department of Software Systems. In 2007, as a consequence of a major reorganization of the University, the number of departments of the Faculty of Computer Science was reduced to one, called the Department of Computer Science.

Over 1000 students are currently studying at the Faculty of Computer Science.
Mission

The mission of the Faculty of Computer Science is to train highly effective specialists, capable of insuring the development of the Information Society in Romania, in the context of our country's integration within the structures of the European Community.

In order to achieve that mission, by promoting excellence in teaching and research activity, the Faculty of Computer Science does the following:

- Offers study programmes for the BSc diploma in Computer Science (both daily courses and distance learning), the diploma of Master in Computer Science, and the PhD diploma in Computer Science. The graduates of the faculty are capable of working as software developers, system engineers, network administrators, information system managers, etc., or as teachers of Computer Science.

- Provides, for each study programme, competitive study plans that allow the recognition of the diplomas in any other country.

- Coordinates the research activity in Computer Science, materialized through the publication of scientific papers in the Technical Report Series of the faculty, in the faculty's review (the Scientific Annals of the University - Computer Science series), and in national and international publications. Participation to scientific events inside and outside Romania, to research projects financed by the Romanian authorities and/or by the European Community, is also encouraged.

- Provides Computer Science courses for other faculties and within the Continuous Education system for the school and high school teachers.

- Participates in European programmes of student and teaching staff exchange.

The Faculty of Computer Science, together with the other faculties, through its entire activity, aims at the development of the local and regional community, at the growth of the "Alexandru Ioan Cuza" University's prestige, both at national and international level.

3. Administrative structure

Board of the Faculty
Dean: prof. dr. Gheorghe Grigoraş
Vice-deans:
- prof. dr. Dan Cristea
- assoc. prof. dr. Sabin Corneliu Buraga
Head of the Department of Computer Science: prof. dr. Dorel Lucanu
Chancellor: lect. dr. Vlad Rădulescu

Administrative staff
Chief administrator of the faculty: Radu Negrescu
Secretaries:
- Chief secretary: Maria Buburuzan
- Study situations: Lăcrămioara Leonte
- Fee problems: Mariana Nichita
- Distance learning: Lavinia Pîrîu
- Teaching problems: Alina Popescu
Phone: +4-0232-201090
Fax: +4-0232-201490
E-mail: secret@infoiasi.ro
Work with students - timetable: Monday to Thursday, 10-12
II. DEGREE PROGRAMMES

1. Qualifications awarded

In accordance to the Bologna system, the duration of BSc studies is 3 years. The graduates of the Faculty of Computer Science achieve the qualification of computer scientists.

The section of Distance Learning has been working since the academic year 2002-2003. The Faculty of Computer Science provides the frame for the continuous improvement of the school and high school teachers.

The MSc (Master) studies have a duration of 2 years. The oldest Master programme in the faculty is on Parallel and Distributed Computing, set up in 1995. Between 1995 and 1998, the section also had a French branch, where the courses were taught by French professors from the Universities of Paris Sud-Orsay, USTL Lille and Sorbonne.

The other Master sections currently functioning in the faculty are: Computational Linguistics (since 2001), Computational Optimization (since 2003), Software Engineering (since 2005), and Information Security (starting in 2009).

2. Admission requirements

Admission to the Faculty of Computer Science is open to all Romanian and EU citizens, as well as for non-EU people. The candidates are required to be high-school graduates and to have studied Mathematics and/or Computer Science in high school.

The admission process consists in the analysis of the personal records of the candidates. The final admission result is computed from the following components:
- the result of the Mathematics/Computer Science test within the high-school graduation exam (50%)
- the final result of the high-school graduation exam (20%)
- the mean of the study results during the high-school study years (30%)

3. Educational and professional goals

The main goal of the Faculty of Computer Science is to teach and train the students in order to become high-performance computer science professionals. This goal is achieved by providing strong theoretical foundations and, at the same time, extensive programming skills. Here are a few domains for which the students qualify after graduation:
- researcher in computer science
- software engineer
- programmer
- database architect/administrator
- system administrator

4. Examination and assessment regulations

The Romanian system comprises marks from 1 to 10. The lowest mark required for passing an exam is 5.

There is a major interest in supporting the continuous evaluation of the students, along with the teaching process. According to the regulations of the "Alexandru Ioan Cuza" University, at least 50% of the final result on each discipline must consist in the assessment of the activity carried out by students during the semester. That is, there are normally two parts of the examination:
- The continuous assessment, during the semester. This may be implemented either by projects that students must fulfill before certain deadlines are reached, or by written/practical tests.
- The final exam, which is sustained within the examination session (at the end of the semester), usually as a written test. The teacher may decide that continuous assessment is sufficient, in which case there is no final exam.
After all parts of the examination are sustained, an overall score is computed for each student. The final result is determined as follows:
- For the scores below the promotion threshold, the final result is between 1 and 4 (failed).
- For the scores above the promotion threshold, the final result is between 5 and 10 and is computed by applying the Gauss distribution over all scores.

5. Final examination

For BSc students, the final examination comprises two components:
- Fundamental knowledge of computer science
- Defending the diploma thesis
The final result is the average of the marks obtained for the two components.
For Master students, the final examination consists in defending the dissertation thesis.
For both BSc and Master students, the evaluation is made by an examination commission. The thesis supervisor is a rightful member of the commission.

6. Access to further studies

BSc graduates may follow a Master specialization, after going through an admission contest. The Master modules of the Faculty of Computer Science are also open for the graduates of other faculties; at the same time, graduates of the Faculty of Computer Science may follow Master modules of other faculties.

Master graduates may continue with PhD studies. The doctoral programme has been included in the structure of the faculty since 1993. There are currently 7 PhD supervisors in the Faculty of Computer Science, each one with his/her own fields of interest.

7. Coordination of teaching activity

ECTS coordinator: assoc. prof. dr. Sabin Corneliu Buraga
Erasmus coordinator: lect. dr. Vlad Rădulescu

8. Study plans

Undergraduate

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>S</td>
<td>L</td>
</tr>
<tr>
<td>Semester I (Year I)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>CS1101</td>
<td>Algorithms and Programming</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CS1102</td>
<td>Computer Architecture and Operating Systems</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>CS1103</td>
<td>Logics for Computer Science</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>CS1104</td>
<td>Mathematics</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>CS1105</td>
<td>Communication in Electronic Environments</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>CS1106</td>
<td>English Language I</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Semester II (Year I)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CS1207</td>
<td>Object-Oriented Programming</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>CS1208</td>
<td>Operating Systems</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>CS1209</td>
<td>Algebraic Foundations of Computer Science</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>CS1210</td>
<td>Probabilities and Statistics</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>CS1211</td>
<td>Hardware Practice</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>CS1212</td>
<td>English Language II</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Semester III (Year II)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CS2101</td>
<td>Computer Networks</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>CS2102</td>
<td>Databases</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>No</td>
<td>Code</td>
<td>Name</td>
<td>Hours per week</td>
<td>Credits</td>
<td>Evaluation</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td>------------------------------------------------</td>
<td>----------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>S</td>
<td>L</td>
</tr>
<tr>
<td>15</td>
<td>CS2103</td>
<td>Formal Languages, Automata and Compilers</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CS2104</td>
<td>Graph Algorithms</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CS2105</td>
<td>Optional package 1</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>CS2106</td>
<td>English Language III</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Optional Disciplines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS2105O1</td>
<td>Coding Theory and Cryptography</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS2105O2</td>
<td>Game Theory</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS2105O3</td>
<td>Continuous Models and Matlab</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>CS2207</td>
<td>Web Technologies</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>CS2208</td>
<td>Advanced Programming Techniques</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>CS2209</td>
<td>Software Engineering</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CS2210</td>
<td>DBMS Practice</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>CS2211</td>
<td>Optional package 2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>CS2212</td>
<td>English Language IV</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Optional Disciplines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS2211O1</td>
<td>Logic Programming</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS2211O2</td>
<td>Functional Programming</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CS3101</td>
<td>Design and Analysis of Algorithms</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>CS3102</td>
<td>Information Security</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>CS3103</td>
<td>Artificial Intelligence</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>28</td>
<td>CS3104</td>
<td>Application Development on .NET Platform</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td>CS3105</td>
<td>Optional package 3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>CS3106</td>
<td>Practice</td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Optional Disciplines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3105O1</td>
<td>Stochastic Processes</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3105O2</td>
<td>Computability, Decidability and Complexity</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>CS3207</td>
<td>Numerical Calculus</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>CS3208</td>
<td>Computer Graphics</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>CS3209</td>
<td>Optional package 4</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>CS3210</td>
<td>Optional package 5</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>CS3211</td>
<td>Optional package 6</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>CS3212</td>
<td>Diploma Paper Work</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Optional Disciplines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3209O1</td>
<td>Rule Based Programming</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3209O2</td>
<td>Bioinformatics</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3210O1</td>
<td>The Modelling of Distributed Systems Using Petri Nets</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3210O2</td>
<td>Genetic Algorithms</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS3211O1</td>
<td>Advanced Operating Systems</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
### Master in Distributed Systems

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>C  S  L  Pr.</td>
<td></td>
<td>P  C  E Mixed</td>
</tr>
<tr>
<td>1</td>
<td>MSD1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MSD1102</td>
<td>Operational Research</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MSD1103</td>
<td>Java Technologies</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MSD1104</td>
<td>Research Project I</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester I (Year I)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSD1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MSD1102</td>
<td>Operational Research</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MSD1103</td>
<td>Java Technologies</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MSD1104</td>
<td>Research Project I</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester II (Year I)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>MSD1205</td>
<td>Advanced Artificial Intelligence</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MSD1206</td>
<td>Parallel Algorithms and Parallel Programming</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MSD1207</td>
<td>Algorithms and Distributed Programming</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MSD1208</td>
<td>Research Project II</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester III (Year II)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>MSD2101</td>
<td>Web Applications Development</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>MSD2102</td>
<td>Distributed Systems Modeling</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MSD2103</td>
<td>Distributed Operating Systems</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MSD2104</td>
<td>Research Project III</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester IV (Year II)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>MSD2206</td>
<td>Project Management</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>MSD2205</td>
<td>Specification and Verification of Concurrent and Distributed Systems</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MSD2207</td>
<td>Security Protocols</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>MSD2208</td>
<td>Dissertation Preparing</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

### Master in Computational Optimization

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>C  S  L  Pr.</td>
<td></td>
<td>P  C  E Mixed</td>
</tr>
<tr>
<td>1</td>
<td>MOC1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MOC1102</td>
<td>Operational Research</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MOC1103</td>
<td>Java Technologies</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MOC1104</td>
<td>Research Project I</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester I (Year I)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MOC1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MOC1102</td>
<td>Operational Research</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MOC1103</td>
<td>Java Technologies</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MOC1104</td>
<td>Research Project I</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester II (Year I)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>MOC1205</td>
<td>Advanced Artificial Intelligence</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MOC1206</td>
<td>Combinatorial Optimization</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MOC1207</td>
<td>Nature Inspired Methods</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MOC1208</td>
<td>Research Project II</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Semester III (Year II)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>MOC2101</td>
<td>Web Applications Development</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Code</td>
<td>Name</td>
<td>Hours per week</td>
<td>Credits</td>
<td>Evaluation</td>
</tr>
<tr>
<td>----</td>
<td>----------</td>
<td>-------------------------------------------</td>
<td>----------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>10</td>
<td>MOC2102</td>
<td>Special Chapters in Artificial Intelligence</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>MOC2103</td>
<td>Experimental Analysis of Algorithms</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>12</td>
<td>MOC2104</td>
<td>Research Project III</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>

**Semester IV (Year II)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>MOC2205</td>
<td>Project Management</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>14</td>
<td>MOC2206</td>
<td>Neural Networks</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>15</td>
<td>MOC2207</td>
<td>Data Mining</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>16</td>
<td>MOC2208</td>
<td>Dissertation Preparing</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>

---

**Master in Software Engineering**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MISS1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>MISS1102</td>
<td>Operational Research</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>MISS1103</td>
<td>Java Technologies</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>MISS1104</td>
<td>Research Project I</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>

**Semester II (Year I)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>MISS1205</td>
<td>Advanced Artificial Intelligence</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>MISS1206</td>
<td>Multimedia Technologies</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>MISS1207</td>
<td>Special Chapters in Human-Computer Interaction</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>MISS1208</td>
<td>Research Project II</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>

**Semester III (Year II)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>MISS2101</td>
<td>Web Applications Development</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>MISS2102</td>
<td>Formal Methods in Software Engineering</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>MISS2103</td>
<td>Software Security</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>12</td>
<td>MISS2104</td>
<td>Research Project III</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>

**Semester IV (Year II)**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>MISS2205</td>
<td>Project Management</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>14</td>
<td>MISS2206</td>
<td>Advanced Databases</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>15</td>
<td>MISS2207</td>
<td>Software Quality</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>16</td>
<td>MISS2208</td>
<td>Work on the Dissertation Thesis</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>

---

**Master in Computational Linguistics**

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ML1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>ML1102</td>
<td>Operational Research</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>ML1103</td>
<td>Java Technologies</td>
<td>2</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>ML1104</td>
<td>Research Project I</td>
<td>4</td>
<td>6</td>
<td>X</td>
</tr>
</tbody>
</table>
### Master in Information Security

<table>
<thead>
<tr>
<th>No</th>
<th>Code</th>
<th>Name</th>
<th>Hours per week</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>C  S  L  Pr.</td>
<td>P  C  E</td>
<td>Mixed</td>
</tr>
<tr>
<td>Semester I (Year I)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MSI1101</td>
<td>Advanced Software Engineering Technics</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MSI1102</td>
<td>Operational Research</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MSI1103</td>
<td>Java Technologies</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MSI1104</td>
<td>Applied Cryptography</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Semester II (Year I)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MSI1205</td>
<td>Models of Security</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MSI1206</td>
<td>Advanced Artificial Intelligence</td>
<td>2  2  6</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MSI1207</td>
<td>Network Security</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MSI1208</td>
<td>Security Protocols</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Semester III (Year II)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>MSI2101</td>
<td>Software Security</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>MSI2102</td>
<td>Wireless and Mobile Security</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MSI2103</td>
<td>Web Applications Development</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MSI2104</td>
<td>Security of Operating Systems</td>
<td>2  3</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>12' MSI2104' Malicious Software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semester IV (Year II)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>MSI2205</td>
<td>Belief Logics in Information Security</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>MSI2206</td>
<td>Security of Electronic Commerce</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MSI2207</td>
<td>Project Management</td>
<td>2  2  8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>MSI2208</td>
<td>Dissertation Preparing</td>
<td>4  6</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
9. Course descriptions

As the Master modules have a series of common courses, only one course description was provided for each such common discipline. Unique course codes are used for these courses:
- MCG1101 for MSD1101, MOC1101, MISS1101, ML1101, and MSI1101
- MCG1102 for MSD1102, MOC1102, MISS1102, ML1102, and MSI1102
- MCG1103 for MSD1103, MOC1103, MISS1103, ML1103, and MSI1103
- MCG1205 for MSD1205, MOC1205, MISS1205, ML1205, and MSI1206
- MCG2101 for MSD2101, MOC2101, MISS2101, ML2101, and MSI2103
- MCG2205 for MSD2205, MOC2205, MISS2205, ML2205, and MSI2207
# ALGORITHMS AND PROGRAMMING

## Course Information

<table>
<thead>
<tr>
<th>Study Year</th>
<th>Semester</th>
<th>Course Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hours Per Week</th>
<th>Total Hours Per Semester</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>C  S  L  Pr.</td>
<td>2  1  1 -</td>
<td>56  94</td>
<td>5</td>
<td>M Romanian</td>
</tr>
</tbody>
</table>

### Objective

- **Algorithms**: Techniques for the design and analysis of efficient algorithms, introduction to a number of fundamental data structures and algorithms (or procedures) for manipulating them, a basic understanding of how common computational problems can be solved efficiently on a computer.
- **Programming**: basic skills in using an imperative programming language (C), basic concepts and techniques of program design, the evaluation of the run time.

### General Description

- **Algorithms**: algorithmic language, static data structures, dynamic data structures, linear lists, trees, graphs (as data structures), heaps, union-find, sorting, searching, problem solving.
- **Programming**: Gradual presentation of the C language (ISO Standard), focusing on the efficient implementation of the fundamental data structures and the case studies presented in the first part.

### Description of Seminary/Laboratory Works

- **Seminar**: problem solving using basic algorithms and fundamental data structures
- **Laboratory**: Fundamental data types and control flow. Files and input-output operations. Arrays and pointers. Functions and macros. Implementation of the fundamental data structures.

### Teaching Methods

- Transparencies and video projector.

### Bibliography

- D. Lucanu: Bazele proiectãrii programelor si algoritmilor, Universitatea "Al. I. Cuza", Iasi, 1996
- Al Kelley, Ira Pohl: A Book on C - Programming in C, Addison Wesley, Reading

### Evaluation

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Seminary Activity (AS), Laboratory Activity (AL), Written Tests (TS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS ≥ 5, AL ≥ 6, TS ≥ 4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation Methods</th>
<th>AS: questions, participation at discussions, original solutions to problems. AL: each topic is evaluated with a mark between 1 and 10. TS: 2 written tests (weeks 7, 13), each test including 8 questions and a problem.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Final Result Formula</th>
<th>Final Mark = 10% AS + 40% AL +50% TS</th>
</tr>
</thead>
</table>
COURSE NAME | COMPUTER ARCHITECTURE AND OPERATING SYSTEMS  
---|---
CODE | CS1102  
STUDY YEAR | 1  
SEMESTER | 1  
COURSE STATUS | (C-compulsory/O optional/F-facultative) | C  

| HOURS PER WEEK | C | S | L | Pr. |  
---|---|---|---|---|  
| TOTAL HOURS PER SEMESTER | 56 |  
| TOTAL HOURS INDIVIDUAL ACTIVITY | 94 |  
| CREDITS | 5 |  
| EVALUATION | (P-during the semester, C-oral examination, E-written examination, M-mixed) | M  
| TEACHING LANGUAGE | Romanian  

| COURSE TEACHERS | TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME | DEPARTMENT  
---|---|---  
| PROF. DR. HENRI LUCHIAN |  
| LECT. DR. VLAD RĂDULESCU | Computer Science  

PREVIOUS COURSES REQUESTED | -  

OBJECTIVES  
At the end of the semester, the students should know:  
- the basic elements of the computer architecture and organization  
- the fundamentals of internal data representation in computers  
- the main functions of an operating system  

GENERAL DESCRIPTION  
Physical representation of information; bits and bytes. Boolean functions; minimization. Logic gates and combinational circuits; adder, decoder. Flip-flops and sequential circuits; counter, shift register.  
Internal data representation. Fixed point representations. Floating point representations.  
The memory; technology, cache memory, virtual memory.  
The structure of the central processing unit. Techniques for performance improvement. Pipeline.  
RISC architecture. Parallel architectures.  
Peripheral devices. The interrupt system.  

DESCRIPTION OF SEMINARY / LABORATORY WORKS  
Boolean functions. Minimization techniques.  
Fixed point representations. Overflow.  
Floating point representations. Overflow.  
Assembly language of the x86 microprocessor family.  

TEACHING METHODS  
Exposition, debate, problem-solving, case studies, exercises.  

BIBLIOGRAPHY (SELECTION)  

EVALUATION  
conditions | The presence at the laboratory and seminary activities.  
---|---

evaluation methods | Two written tests: the first (TS1) during the 7th week (40 hours individual study), the second (TS2) during the examination session (40 hours individual study). One practical test (TL) during the 14th week, regarding the laboratory activity (14 hours individual study).  
final result - formula | If at least 5 points were achieved on each of the three tests:  
Final result = (TS1 + TS2 + TL) / 3  
If 4 points were achieved on one of the tests:  
Final result = (TS1 + TS2 + TL) / 3, with truncation  
If 3 points were achieved on one of the tests:  
Final result = (TS1 + TS2 + TL) / 3 - 1, with truncation  

criteria | At least 5 points on each test; 5 or 4 points are accepted for at most one test.  

13
**Course Name**: LOGICS FOR COMPUTER SCIENCE  
**Code**: CS1103

<table>
<thead>
<tr>
<th>Study Year</th>
<th>Semester</th>
<th>Course Status (C-compulsory/OP-optional/F-facultative)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hours per Week</th>
<th>Total Hours Per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

**Course Teacher**: PROF. DR. CRISTIAN-DUMITRU MASALAGIU  
**Department**: Computer Science

**Previous Courses Requested**: -

**Objectives**
1. Getting minimal skills for using Logic in Computer Science and understanding its importance
2. Understanding the needed fundamental concepts for the future study of related disciplines such as: Logic Programming, Specification and Verification of Real Systems, Expert Systems Web Ontologies or Non-classical Logics
3. Getting minimal skills to rigorously work with software applications and media

**General Description**
1. Logic in Computer Science (introduction, motivation)
2. Boolean Algebras (semantic domains)
3. Propositional Logic (LP)
4. First-order Predicate Calculus (LP1)
5. Introduction to Deductive Systems and Logical Theories
6. Introduction to Logic Programming
7. The Idea of Verification

**Description of Seminary/Laboratory Works**
The main goal of any Seminar is to facilitate a deeper understanding of the content of the previous Courses, with the help of more complicated, new and detailed examples. This will be accomplished with the direct participation of the students.

**Teaching Methods**
All the classical didactic methods will be used: systematic exposure of knowledge, conversation, learning „by discovery”, etc. The Courses will be taught using a video-projector.

**Main Bibliography (Selection)**
3. Specific INTERNET sites

**Evaluation**
- **Conditions**: Every student will be tested 1 to 4 times during the 14-th seminaries (see bellow), not necessarily immediately to the corresponding course. A bonus may be granted for supplementary (good) answers.
- **Criteria**: The above examinations can produce a maximum of 60 points. The exam (in the special session of the 7th and 14th week) may produce other (maximum) 60 points. To “graduate” the course, a minimum of 40 points is needed.
- **Evaluation Methods**: Additional written test may be given at fixed or no-announced dates. Thus, the total amount of points needed to pass is between 40 and 120.
- **Final Result Formula**: The final grade is computed by first summing up all the obtained points and then by dividing the result by ten (between 40 to 50 points the degree will be 5.00). The grades will be then rounded such as to get a Gauss curve for the given year of study (see the regulations). The grades greater than 10 will be rounded to ten.
### Objectives

- To systematize and to get thoroughly into the study of some theoretical and applied questions on differential and integral calculus in the spaces $\mathbb{R}$, $\overline{\mathbb{R}}$, and $\mathbb{R}^n$ ($n \geq 2$). To present certain basic results and applications in the domain of real, differentiable and integrable, scalar and vectorial functions.
- To point out fundamental ideas in real analysis, algebra and geometry of curves and surfaces for the benefit of other objects of study at the Faculty of Computer Science.

### General Description

Elements of algebra, analysis (topology) and geometry in connection with the spaces $\mathbb{R}$, $\overline{\mathbb{R}}$, and $\mathbb{R}^n$ ($n \geq 2$). Numerical sequences and series. Functions of one and several real variables (generalities, limits, continuity, derivatives, differentials, series and integrals). Basic applications.

### Description of Seminary Works

Sets, relations and functions (generalities). Basic algebraical structures. Numerical real sets. Remarkable numerical inequalities. Sequences and series of real numbers. Algebraic and topological aspects of the space $\mathbb{R}^n$ ($n \in \mathbb{N}^+$). Real functions (generalities). Linear, affine and quadratic real forms (algebraic considerations and geometrical interpretations). Limits and continuity for real functions. Derivatives, differentials and its applications (implicit functions, local inverse functions, functional dependence or independence, unconditioned and conditioned extreme problems). Sequences and series of real functions (series of powers, Taylor and Fourier series). Simple and multiple, definite and indefinite, without or with parameters Riemann integrals.

### Teaching Methods


### BIBLIOGRAPHY (SELECTION)

- F. Iacob – *Mathematics* (Romanian Course and Seminary Notes; on the mentioned site), 2008.

### Evaluation

- **Conditions**: Compulsory participation at the written examinations during the semester and the exam.
- **Criteria**: Presence and activity at the seminars. Home work achievement. Written examination results. Participation in counselling programme.
- **Evaluation Methods**: Appreciation by marks on the frequency (NFS1,NFS2) and activity at the seminars (NPS1,NPS2), homeworks (NT1,NT2), written examinations (NL1,NL2) and participation in counselling programme (NPC1, NPC2).
- **Final Result-Formula and the Pass Mark**:
  
  \[ PF = (0.75 \times \text{NFS1} + \text{NPS1} + 1.25 \times \text{NET1} + 1.75 \times \text{NL1} + 0.25 \times \text{NPC1}) + (0.75 \times \text{NFS2} + \text{NPS2} + 1.25 \times \text{NET2} + 1.75 \times \text{NL2} + 0.25 \times \text{NPC2}) \geq 45; \]
  
  The final mark (NF) is established from PF, in accordance with the new ECTS norms.
## Communication in Electronic Environments

**Course Code:** CS1105

<table>
<thead>
<tr>
<th>Study Year</th>
<th>Semester</th>
<th>Course Status (C-compulsory/OP-optional/F-facultative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>C</td>
</tr>
</tbody>
</table>

### Hours per Week

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>Total Hours per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation (P: during the semester, C: oral examination, E: written examination, M: mixed)</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>42</td>
<td>108</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

### Previous Courses Requested

- 

### Objectives

- Development of the abilities for structuring a spoken / written discourse, in different communication contexts and situations. How to organize and present the ideas, how to format the documents.
- Understanding of the Internet functioning mode, and usage of its main services.
- Web sites development, by using HTML and CSS according Web design principles.

### General Description

- XHTML - eXtensible HyperText Markup Language
- CSS - Cascading Style Sheets
- Web resources search and retrieval
- Web sites design and development
- Social Web (Web 2.0)
- Electronic documents. Microformats
- On-line communities

### Description of Seminary / Laboratory Works

- Thematic Web sites design, and case studies for different Web sites types: presentation Web sites, e-commerce, e-learning, web portals
- Tools for Web sites design and development
- Using various social Web applications. Integrating in different on-line communities
- MS Office: using and defining of various documents types templates. Document formatting, text processing
- Organizing, conceiving, and presentation of different types of discourses, papers, writings, documents: CV, tutorial, degree thesis, book/product/company presentation

### Teaching Methods

- Tutorial exposition, case studies, dialogue.

### Bibliography

2. C. Bertrand (coord.), O introducere in presa, Polirom, 2001
3. M. Brut, Instrumente pentru E-Learning, Polirom, 2006
4. S. Buraga, Proiectarea siturilor Web (editia a II-a), Polirom, 2005
6. L. Goin, Design for Web Developers: Colour and Layout for the Artistically Overwhelmed, DMXzone.com, 2005
8. R. Hoff, Regulile unei prezentari de succes, Curtea Veche, 2002
10. S. Prutianu, Antrenamentul abilitatilor de comunicare, Ed. Polirom, Iasi, 2004

### Evaluation

- Conditions: Minimum 50 points for each of the two projects
- Criteria: Spoken / written discourse structuring, complying with the design principles, originality
- Evaluation Methods: Project, activity in practical works, systematic involving observation
- Final Result Formula: 
  \[ PF = 0.4 \times P1 + 0.4 \times P2 + 0.1 \times AL + 10 \]
  where \( PF \) = final score, \( P1 \) = score for project 1, \( P2 \) = score for project 2, \( AL \) = score for practical works activity. The final mark will be established according the Gauss diagram.
# ENGLISH

<table>
<thead>
<tr>
<th>COURSE NAME</th>
<th>ENGLISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>CS1106</td>
</tr>
</tbody>
</table>

## Study Year 1, Semester 1

### Course Status
- C: Compulsory

### Hours Per Week
- C: Contact Hours
- S: Seminar Hours
- L: Laboratory Hours
- Pr: Practical Hours

<table>
<thead>
<tr>
<th>Hours Per Week</th>
<th>Total Hours Per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 2 0</td>
<td>28</td>
<td>122</td>
<td>5</td>
<td>M</td>
<td>ENGLISH</td>
</tr>
</tbody>
</table>

### Teacher
- DRD. DRAGOS ZETU
- Computer Science

## Previous Courses Requested

### Objectives
To improve the student’s proficiency in English in general, computer science English in particular.

### General Description
The seminars will provide the student the possibility to work with authentic materials, interact and improve his/her grammar skills.

### Description of Seminar / Laboratory Works
The seminars will provide the student the possibility to work with authentic materials, interact and improve his/her grammar skills.

### Teaching Methods

### Bibliography (Selection)
Any English grammar compendium

### Evaluation
- **Conditions**: Attendance, written exam
- **Criteria**: Good proficiency in English
- **Evaluation Methods**: Continuous evaluation, written exam
- **Final Result Formula**: 50% exam results, 50% seminar activity
## OBJECT ORIENTED PROGRAMMING

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>III</th>
<th>SEMESTER</th>
<th>1</th>
<th>COURSE STATUS (C-compulsory/OP-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. DOREL LUCANU</td>
<td></td>
<td>Computer Science</td>
</tr>
<tr>
<td>PROF. DR. GHEORGHE GRIGORAS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
<th>Algorithms and Programming (CS1101)</th>
</tr>
</thead>
</table>

**OBJECTIVES**

Teaching fundamental concepts in object-oriented programming and the use of the programming language C++.

**GENERAL DESCRIPTION**

Language C++ (ISO Standard) with emphasis on the representation of objects and classes in C++, the relationship of derivation and description of hierarchies of classes in C++, virtual functions and implementation of polymorphism in C++, templates and parametric classes, using standard library. Concepts and principles in Object Oriented Programming: classes, objects, hierarchies of class, polymorphism, abstract classes, interfaces, parametric classes, observer pattern, composite pattern, iterator pattern, case studies.

**DESCRIPTION OF SEMINARY / LABORATORY WORKS**

Introduction classes in C++, shifting from C to C++, advanced classes, inheritance, polymorphism, templates, parametric classes, STL: Standard Template Library.

**TEACHING METHODS**

Slides with course items; seminar themes; projects’ issues; electronic version of the course; main readings will be find on the web page.

**BIBLIOGRAPHY**

(Selection)

- Stanley B. Lippman: C++ Primer. Addison Wesely, 1992

**EVALUATION**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Participation to laboratory hours (LA), participation to writing tests (WT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>LA &gt;= 6, WT &gt;= 4</td>
</tr>
<tr>
<td>Evaluation methods</td>
<td>Mixed (during the semester and examination)</td>
</tr>
<tr>
<td>Final result – formula</td>
<td>40% LA + 60% WT</td>
</tr>
</tbody>
</table>
COURSE NAME: OPERATING SYSTEMS

STUDY YEAR: 1  SEMESTER: 2  COURSE STATUS: C- compulsory/OP- optional/ F-facultative

HOURS PER WEEK

<table>
<thead>
<tr>
<th>Credits</th>
<th>Total Hours of Individual Activity</th>
<th>Total Hours per Semester</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>56</td>
<td>94</td>
<td>5</td>
<td>P</td>
</tr>
</tbody>
</table>

TEACHING LANGUAGE: Romanian

TAUGHT BY

Lect. Dr. Cristian Vidraşcu

DEPARTMENT: Computer Science

REQUIRED COURSES

Computer Architecture and Operating Systems, C Programming

OBJECTIVES

The students who will attend this course will obtain knowledge about operating systems, regarding the techniques used for the design and the implementation of them. Also, they will obtain the skills to write parallel processing programs and to use the UNIX/Linux operating system.

GENERAL THEMATICS


SEMINARY/LABORATORY THEMATICS


TEACHING METHODS

Exposure using video-projector, demos on blackboard and on computer.

BIBLIOGRAPHY


EVALUATION

Evaluation conditions: The presence at the laboratory activities and at the written theses.

Evaluation criteria: Minimal score for graduation: TS1 + TS2 ≥ 25p . L ≥ 10p

Evaluation modes: Evaluation during the semester: lab works and two written theses.

Evaluation formula: Final Score = Lab * 0.4 + WrittenThesis1 * 0.3 + WrittenThesis2 * 0.3. The final mark is obtained from the final score through classification based on the ECTS – European Credit Transfer System and Diploma Supplement.
<table>
<thead>
<tr>
<th>COURSE NAME</th>
<th>ALGEBRAIC FOUNDATIONS OF COMPUTER SCIENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE:</td>
<td>CS1209</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/O-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-coral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>56</td>
<td>94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. FERUCIO LAURENTIU TIPLEA</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
<th>No prerequisite required.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
<th>The course deals with those topics from mathematics that have proven to be particularly relevant to students in computer science.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>GENERAL DESCRIPTION</th>
<th>The course covers basic elements on set theory (sets, relations and functions, induction and recursion), partially ordered sets (posets, lattices), number theory (divisibility, prime numbers, congruences), semigroups and monoids, groups, rings, vector spaces and linear mappings. Relevant applications of each chapter of the course to computer science, are discussed.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DESCRIPTION OF SEMINARY / LABORATORY WORK</th>
<th>Seminars and laboratories are grouped around the chapter currently discussed in the course. They are aimed to illustrate the topics of the chapter mainly by practical applications.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TEACHING METHODS</th>
<th>On-line and blackboard presentation.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>BIBLIOGRAPHY (SELECTION)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>EVALUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>conditions</td>
</tr>
<tr>
<td>criteria</td>
</tr>
<tr>
<td>evaluation methods</td>
</tr>
<tr>
<td>final result - formula</td>
</tr>
</tbody>
</table>
## Course Details

<table>
<thead>
<tr>
<th>Course Name</th>
<th>Probabilities and Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>CS1210</td>
</tr>
<tr>
<td>Study Year</td>
<td>I</td>
</tr>
<tr>
<td>Semester</td>
<td>2</td>
</tr>
<tr>
<td>Course Status</td>
<td>C (compulsory)</td>
</tr>
</tbody>
</table>

### Hours Per Week

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>Total Hours Per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

### Course Teacher

<table>
<thead>
<tr>
<th>Teaching and Scientific Degree, First Name, Last Name</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assoc. Prof. Dr. Silvia Luchian</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

### Previous Courses Requested

Communication in Electronic Environments (MS Office).

### Objectives

Students should be able to apply fundamental probabilistic models and methods to solve problems related to the study of random phenomena; they should also be able to use the computer in order to apply statistical methods for decision making, based on experimental data. Students should be able to recognize various types of random variables and to understand which variable can be used in given situations; they should also understand the basics of statistical reasoning, specifically to select and apply appropriate statistical tests for hypothesis testing.

### General Description

**Descriptive Statistics**: synthesis and presentation of experimental data (see seminar below).


**Inferential Statistics**: Parameter estimation. Confidence intervals for population parameters. Hypothesis testing for means, proportions, dispersions. Inferences on multinomial experiments. Inferences over two populations. Dispensational analysis.

### Description of Seminary/Laboratory Works

Students will practice, using EXCEL®, how to solve specific real world problems using notions from Probability Theory, as well as various methods of Descriptive Statistics for organising and presenting raw data (relative and cumulative frequencies; proportions; frequency distributions; graphical representation of random variables; measures of central tendency; measures of variation), as well as raw data processing for statistical analysis: confidence intervals for means, proportions, dispersions; significance tests for means, proportions, dispersions, including non-normal populations; inferences for two populations; qualitative variables; the Chi-square test (independence, homogeneity); dispensational analysis.

### Teaching Methods

Exposition, problem-solving, case studies, exercise.

### Bibliography

- Ciucu, Gh., Craiu, V.: Introducere în Teoria probabilităŃilor şi Statistică matematică, Editura Didactică şti Pedagogică.
- Ciucu, Gh., Craiu, V., Sâcuiu, I.: Probleme de Teoria probabilităŃilor, Editura Tehnică.

### Evaluation

- **Conditions**: results corresponding to at least 80% of seminar/laboratory contact hours.
- **Criteria**: For passing, at least a 6 grade is required for the seminar/laboratory tests.
- **Evaluation Methods**: Three tests: two tests for seminar/laboratory hours (during the semester; about 4 hours per week expected workload) and one final test from the course content (during examination weeks; about 1.5 hours per week expected workload during teaching period and 14 hours recap during examination period). In total, about 94 hours expected workload.
- **Final Result - Formula**: Weighted average between the average mark of the two practical tests (weight 50%) and the mark for the final test (weight 50%).
**COURSE NAME**
HARDWARE PRACTICE

**CODE:** CS1211

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/O-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

**HOURS PER WEEK**
Total hours per semester

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>42</td>
<td>108</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

**COURSE TEACHER**

TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME
ENG. ADRIAN BUBURUZAN
LECT. DR. VLAD RĂDULESCU

DEPARTMENT
Computer Science

**PREVIOUS COURSES REQUESTED**
Computer Architecture and Operating Systems

**OBJECTIVES**
Understanding the hardware structure of the computers. Acquiring the knowledge regarding the hardware and software configuration of a PC.

**GENERAL DESCRIPTION**

**DESCRIPTION OF SEMINAR / LABORATORY WORKS**
Assembling and configuring a computer, working with hard disks, building and configuring computer networks.
Low-level programming.

**TEACHING METHODS**
Exposition, practical demonstrations.

**BIBLIOGRAPHY**
(Selection)

**EVALUATION**
Final result formula
Final result = (TS + TL1 + TL2) / 3

<table>
<thead>
<tr>
<th>conditions</th>
<th>criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>The presence at the laboratory activities.</td>
<td></td>
</tr>
<tr>
<td>At least 5 points on each test; 4 points are accepted for at most one test.</td>
<td></td>
</tr>
</tbody>
</table>

| evaluation methods | Two practical tests, the first (TS1) during the 7th week, the second (TS2) during the 14th week. One written test (TL) during the examination session. |

<p>| final result - formula | Final result = (TS + TL1 + TL2) / 3 |</p>
<table>
<thead>
<tr>
<th>COURSE NAME</th>
<th>ENGLISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE: CS1212</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>1</th>
<th>SEMESTER</th>
<th>2</th>
<th>COURSE STATUS (C-compulsory/OP-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOURS PER WEEK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL HOURS PER SEMESTER</td>
<td>28</td>
<td>122</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREDITS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVALUATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEACHING LANGUAGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENGLISH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRD. DRAGOS ZETU</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
<th>To improve the student’s proficiency in English in general, computer science English in particular.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERAL DESCRIPTION</td>
<td>The seminars will provide the student the possibility to work with authentic materials, interact and improve his/her grammar skills.</td>
</tr>
<tr>
<td>DESCRIPTION OF SEMINARY / LABORATORY WORKS</td>
<td>The seminars will provide the student the possibility to work with authentic materials, interact and improve his/her grammar skills.</td>
</tr>
<tr>
<td>TEACHING METHODS</td>
<td></td>
</tr>
</tbody>
</table>

| BIBLIOGRAPHY (SELECTION) | Any English grammar compendium. |

<table>
<thead>
<tr>
<th>EVALUATION</th>
<th>conditions</th>
<th>Attendance, written exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>Good proficiency in English</td>
<td></td>
</tr>
<tr>
<td>evaluation methods</td>
<td>Continuous evaluation, written exam</td>
<td></td>
</tr>
<tr>
<td>final result - formula</td>
<td>50% exam results, 50% seminar activity</td>
<td></td>
</tr>
</tbody>
</table>
# COMPUTER NETWORKS

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>II</th>
<th>SEMESTER</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE STATUS</td>
<td>(C- compulsory, O- optional, F- facultative)</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>56</td>
<td>124</td>
</tr>
</tbody>
</table>

### COURSE TEACHER

<table>
<thead>
<tr>
<th>ASSOC. PROF. DR. SABIN-CORNELIU BURAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science</td>
</tr>
</tbody>
</table>

### PREVIOUS COURSES REQUESTED

Operating Systems, Algorithms and Programming

### OBJECTIVES

To provide a comprehensible vision of computer networks design issues. The students will be able to program network (Internet) applications based on TCP/IP protocols. Key information regarding different Internet paradigms will be also provided: client/server model, remote procedure call, and peer-to-peer model.

### GENERAL DESCRIPTION


### SEMINAR/LABORATORY WORKS


### TEACHING METHODS

Interactive presentations. Direct interaction. Online access to additional resources via the Website course.

### BIBLIOGRAPHY


### EVALUATION

<table>
<thead>
<tr>
<th>conditions</th>
<th>1 project (P), 1 optional test during semester (T), lab assignments (L), other individual activities (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>project P &gt; 5, test T &gt; 5</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>1 project (P), 1 optional test during semester (T), lab assignments (L), other individual activities (A)</td>
</tr>
<tr>
<td>final result-formula</td>
<td>0.4<em>P+0.3</em>1+0.1<em>L+0.1</em>A+1</td>
</tr>
</tbody>
</table>
**COURSE NAME**: DATABASES  
**CODE**: CS2102

<table>
<thead>
<tr>
<th><strong>STUDY YEAR</strong></th>
<th><strong>SEMESTER</strong></th>
<th><strong>COURSE STATUS (C-compulsory/O-optimal/F-facultative)</strong></th>
<th><strong>COURSE STATUS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>1</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>HOURS PER WEEK</strong></th>
<th><strong>TOTAL HOURS PER SEMESTER</strong></th>
<th><strong>TOTAL HOURS INDIVIDUAL ACTIVITY</strong></th>
<th><strong>CREDITS</strong></th>
<th><strong>EVALUATION</strong></th>
<th><strong>TEACHING LANGUAGE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L. Pr.</td>
<td>56</td>
<td>94</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</strong></th>
<th><strong>DEPARTMENT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. VICTOR FELEA</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

**PREVIOUS COURSES REQUESTED**: -

**OBJECTIVES**: The presentation of some theoretical models for databases: relational, entity-relational. The implementation of these models and the realization of programming products in ORACLE.


**TEACHING METHODS**: Presentation of the course content., verification of individual themes realized in Oracle.

**BIBLIOGRAPHY (SELECTION)**

**EVALUATION**

<table>
<thead>
<tr>
<th><strong>conditions</strong></th>
<th><strong>Laboratory Activity (Lab), Written Tests (L1, L2)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>L1+L2&gt;=20, Lab &gt;=20</td>
</tr>
</tbody>
</table>

**evaluation methods**
- L1 in 7-th week, L2 in 14-th week
- Lab in laboratories.

**final result - formula**: The marks depend on the percentages fixed by the university.
COURSE NAME: FORMAL LANGUAGES, AUTOMATA AND COMPILERS  
CODE: CS2103

STUDY YEAR: III  
SEMESTER: 1  
COURSE STATUS (C-compulsory/OP-optional/F-facultative): C

HOURS PER WEEK  
TOTAL HOURS PER SEMESTER  
TOTAL HOURS INDIVIDUAL ACTIVITY  
CREDITS  
EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)  
TEACHING LANGUAGE

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>Total</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>M Romanian</td>
</tr>
</tbody>
</table>

COURSE TEACHER  
PROF. DR. GHEORGHE GRIGORAS  
DEPARTMENT  
Computer Science

PREVIOUS COURSES REQUESTED  
Algorithms and Programming (CS1101), Object-Oriented Programming (CS1207)

OBJECTIVES  
Teaching fundamental concepts and results on the formal languages (especially those of type 2 and 3), finite automata and pushdown automata.

Building a lexical analyzer using regular expressions and a scanner generator e.g. Lex

Building a syntactic analyzer for a context free grammar using a parser generator e.g. Yacc

Using Lex - Yacc for designing an interpreter/compiler for a programming language

GENERAL DESCRIPTION  
Languages and grammars, grammars classification (Chomsky hierarchy), Regular grammars and languages, Finite automata and accepted languages, Equivalence of deterministic models with the nondeterministic ones and with regular grammars, Context free grammars and languages, Derivations in context free grammars, Ambiguity, Recognition of context free languages, Removing epsilon rules, Removing rules of the form A->B, Chomsky normal form, Pushdown automata, Programming languages: design, implementation, Lexical analysis, Syntax analysis, Semantic analysis, Top down (predictive) syntax analysis, Bottom up (shift reduce) syntax analysis, Recursive descent syntax analysis, LL Syntax Analysis, LR Syntax Analysis, Translation to intermediate code.

DESCRIPTION OF SEMINARY / LABORATORY WORKS  
Examples of languages and grammars, Deterministic finite automata, Nondeterministic automata, Epsilon-transition automata; example, Regular expressions, examples with reference to lexical units of programming languages, Context free grammars, the derivation of trees, eliminating unnecessary symbols, eliminating rules of erasing, Chomsky normal form, with word recognition algorithm CYK, Automatic Pushdown; examples, lexical analyzer’s manual, obtained with an analyzer tool LEX, analyze using tools YACC type. Interpreter built with Lex and YACC.

TEACHING METHODS  
Slides with course items; seminar themes; projects’ issues; electronic version of the course; main readings will be find on the web page

BIBLIOGRAPHY (SELECTION)  
Stoughton Alley, Formal Language Theory, Kansas State University, Draft of Fall 2007.
Yehezkael R.B., Course notes on Formal Languages and Compilers, Jerusalem College of Technology, December 2004.

EVALUATION  
Conditions  
Seminars’ activity (SA), participation to tutorial hours for clarifying the issues regarding project elaboration, participation to laborator hours (LA), participation to final exam (FE)

Criteria  
SA >= 5, LA >= 5, FE >= 5

Evaluation methods  
Mixed (during the semester and examination)

Final result – formula  
Formula of the final score: 30% SA + 30% LA + 40% FE and ECTS criteria
### COURSE NAME
GRAPH ALGORITHMS

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEMESTER</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>CS2104</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE STATUS (C- compulsory/ O- optional/ F- facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P- during the semester, C- oral examination, E- written examination, M- mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>56</td>
<td>124</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. CORNELIUS CROITORU</td>
<td>Computer Science</td>
<td></td>
</tr>
</tbody>
</table>

| PREVIOUS COURSES REQUESTED | Algorithms and Programming (data structures) (CS1101) |

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
<th>The students will be familiarised with the basic notions and results of the Algorithmic Graph Theory, which will be applied in the design of efficient algorithms for various combinatorial optimization problems.</th>
</tr>
</thead>
</table>

| GENERAL DESCRIPTION | Complexity Classes, Graph Theory vocabulary, Path problems (graph traversal, shortest paths, connectivity), Minimum spanning trees (union-find, amortized complexity), Matchings, Flows, Polinomial reductions for decision problems on graphs, Approaches for NP-hard problems on graphs, Planar Graphs. |

| DESCRIPTION OF SEMINARY/ LABORATORY WORKS | Each seminar debates 4 problems (some of them, very difficult) in order to deepen the subjects introduced in the course. All problems are posted at the beginning of the semester such that interested students could try to find original solutions or to search similar questions in the related bibliography. |

| TEACHING METHODS | Video presentations of the slides (containing the course notes) available in pdf format at the beginning of the semester (http://thor.info.uaic.ro/~croitoru/ag/ag08-09allinone.pdf). |

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOMESCU I., Probleme de combinatorica si teoria grafurilor, Editura did. si ped., Bucuresti, 1981.</td>
</tr>
<tr>
<td></td>
<td>DIESTEL R., Graph Theory, Electronic Edition.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVALUATION</th>
<th>conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>A student will be considered to have passed the exam if (s)he obtains at least 50 points.</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>- Seminary activity (attendance, work quality): 0-18 points.</td>
</tr>
<tr>
<td></td>
<td>- Homeworks (3 homeworks, in weeks 4, 8, 12) each giving maximum 14 points: 0-42 points.</td>
</tr>
<tr>
<td></td>
<td>- Written Final test: 0-60 points.</td>
</tr>
<tr>
<td>final result - formula</td>
<td>The final grade (if the total number of points is at least 50) is given by applying the ECTS rules (adapted to FIU).</td>
</tr>
</tbody>
</table>
The minimal prerequisites for students taking this course are a rather elementary knowledge of linear algebra, number theory, algorithm design, and complexity. Students with more mathematical background and maturity will be able to move rather quickly through the material.

The course is designed to introduce coding theory and cryptography to students in computer science, mathematics, and engineering.

The course is divided into two parts. The first part, coding theory, deals with variable length codes with application to data compression, and block codes with application to data communication over a noisy channel. The second part, cryptography, introduces several major themes in the field: perfect secret, symmetric-key cryptography, public-key cryptography, hash functions, digital signatures, authentication, secret sharing, application to computer security (Kerberos, PGP, IP-security, electronic commerce etc.).

Seminars and laboratories are grouped around the chapter currently discussed in the course. They are aimed to illustrate the topics of the chapter mainly by practical applications.

On-line and blackboard presentation.

• F.L. Tiplea. Introduction to Coding Theory (in preparation).
• F.L. Tiplea. Introduction to Cryptography (in preparation).

conditions

criteria

evaluation methods 6 homeworks and a final exam

final result - formula 50% from the homeworks and 50% from the final exam
COURSE NAME | CONTINUOUS MODELS AND MATLAB | CODE: CS2105O2
---|---|---
STUDY YEAR | II | SEMESTER | 1 | COURSE STATUS ( C-compulsory / OP-optional / F-facultative ) | OP

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-coral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

COURSE TEACHER |
LECT. DR. FLORIN IACOB | DEPARTMENT | Computer Science |

PREVIOUS COURSES REQUESTED |
Mathematics; Algorithms and Computer Programming; e-Communication; English (I + II); Oriented - Object Programming.

OBJECTIVES |
- The presentation of the basic concepts of mathematical modelling, mainly relying on differential equations and dynamical systems.
- The knowledge about MATLAB and its applications in the study of continuous mathematical models.

GENERAL DESCRIPTION |
Mathematical modelling (generalities, basic notions and principles). Discrete and continuous models (description, main characteristics). Differential equations and dynamical systems (basis). Some models governed by differential equations (examples). MATLAB (foundations; important tools; programming environment). The analysis of continuous models by MATLAB facilities.

DESCRIPTION OF SEMINARY / LABORATORY WORKS |

TEACHING METHODS |

BIBLIOGRAPHY (SELECTION) |

EVALUATION |
conditions | Full participation at the laboratory works.

criteria | Course and seminar/laboratory frequency; Active participation at seminars and laboratory classes; Homeworks preparation; Running the written tests during the semester and the exam.

evaluation methods | Marking by points the frequency at course (PCF), seminar (PSF) and laboratory (PLF) classes, the activity at seminars (PAS) and laboratory works (PALW), the homeworks effectuation (PHE), the involving in the counselling programme (PICP) and the result at the two tests (TS1, TS2).

final score (FS) formula, examination condition; final mark establishing | $FS = 60\% \times PTS + 20\% \times TS1 + 20\% \times TS2$, where $PTS = PCF + PSF + PAS + PLF + PALW + PHE + PICP \geq 30$.

Final mark is established using FS, by the new rule ECTS.
OBJECTIVES
This course is intended to provide a general insight in the field of game theory and its applications in real-life situations, economics, social sciences, computer science, operations research, etc. Game theory deals with mathematical models for competition and cooperation. The course is mainly aimed to enlighten the benefits achieved via interactions between game theory and computer science. This course is a must for students interested in (Master and) PhD programs and scientific research concerning (the interface between informatics and) game theory.

GENERAL DESCRIPTION

DESCRIPTION OF SEMINARY/LABORATORY WORKS
Representing a broad range of real-life situations (including computer science situations) and parlor games (including games that computers play) as non-cooperative or cooperative games. Solving different non-cooperative and cooperative games using traditional methods and available software. Analysing the complexity of algorithms for computing solutions of games. Designing algorithms for computing solutions of special classes of cooperative games.

TEACHING METHODS
Using overhead projector and blackboard.

BIBLIOGRAPHY (SELECTION)

EVALUATION
| conditions | ME (midterm evaluation), FE (final evaluation), W (Seminary/Laboratory Works) |
| criteria  | ME ≥ 6, FE ≥ 4, W ∈ {0, 1, 2} |
| evaluation methods | ME (written test (75 min) covers weeks 1-6) and FT (written test (75 min) covers weeks 8-13) |
| final result - formula | ME + FE + W |
## OBJECTIVES
Students will learn to use a language adequate to the Computer Science field. Students will gain the necessary skills to participate in a job interview, will be able to write a CV and a covering letter in English, etc.

## GENERAL DESCRIPTION OF THE COURSE
Grammar: “if” clauses, adjectives, sequence of tenses, prepositions. Vocabulary: specific to Computer Science + other fields of interest for the students. Topics of discussion: Artificial Intelligence, Cryptography, Windows vs Linux, Dependence on computers, etc.

## DESCRIPTION OF SEMINARY / LABORATORY WORKS
The students will present a project in order to prove their ability of using accurate English and of expressing their ideas by means of a vocabulary adequate to the themes they present.

## TEACHING METHODS
Interactive methods used all along the seminar.

## EVALUATION
<table>
<thead>
<tr>
<th>conditions</th>
<th>criteria</th>
</tr>
</thead>
</table>
| Var 1.: Seminar activity  
Project presentation  
Written exam | Var 1.: Seminar minimum score: 50% of the maximum score that can be obtained at the seminar  
(minimum 50% of the score for the seminar homework and minimum 50% of the score for the project)  
Var 2.: Seminar minimum score: 50% of the maximum score that can be obtained at the seminar  
(minimum 50% of the score for the seminar homework and minimum 50% of the score for the project)  
Exam minimum score: 40% of the score that can be obtained at the exam |
| evaluation methods | Seminar: seminar homework, project  
Written exam– work time: 1 hour |

The final result is:
Var 1.: a sum of the scores obtained for the seminar and for the project  
Var 2.: a sum of the scores obtained for the seminar, for the project and for the exam.

The student that takes the exam will receive a grade, otherwise he/she will be considered absent. If one of the criterion of promotion is not fulfilled, the student will receive a grade equal to or lower than 4.
**COURSE NAME**: WEB TECHNOLOGIES  
**CODE**: CS2207

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>56</td>
</tr>
</tbody>
</table>

**COURSE TEACHER**: ASSOC. PROF. DR. SABIN-CORNELIU BURAGA  
**DEPARTMENT**: Computer Science

**PREVIOUS COURSES REQUESTED**: Computer Networks, Formal Languages, Automata and Compilers, Algorithms and Programming

**OBJECTIVES**: To study the actual methodologies, specifications and techniques of Web development and to offer the necessary skills for designing, implementing and deploying complex Web applications.

**GENERAL DESCRIPTION**: General architecture of the WWW space. Uniform resource identifiers (URIs). Hypertext. HTTP protocol.


Web services: SOAP, WSDL, UDDI. REST paradigm. Service oriented architecture (SOA).


Introduction to Web engineering.

Web resource searching and retrieval technologies: robots, search engines, other approaches.

Web application security.


**TEACHING METHODS**: Interactive presentations. Direct interaction. Online access to additional resources via the Website course.

**BIBLIOGRAPHY**  
5. S. Buraga (coord.), Programarea in Web 2.0 (in Romanian), Polirom, 2007.

**EVALUATION**:  
- **conditions**: 1 project (P), 1 optional test during semester (T), lab assignments (L), other individual activities (A)  
- **criteria**: project P>5, test T>5

- **evaluation methods**: 1 project (P), 1 optional test during semester (T), lab assignments (L), other individual activities (A)

- **final result - formula**: $0.4P+0.3T+0.1L+0.1A+1$
COURSE NAME: ADVANCED TECHNIQUES OF PROGRAMMING  
CODE: CS2208

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/OP-optional/F-facultative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>2</td>
<td>C</td>
</tr>
</tbody>
</table>

HOURS PER WEEK    | TOTAL HOURS PER SEMESTER | TOTAL HOURS INDIVIDUAL ACTIVITY | CREDITS | EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed) | TEACHING LANGUAGE |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>56</td>
<td>154</td>
<td>5</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

COURSE TEACHER: LECT. DR. CRISTIAN FRĂSINARU  
DEPARTMENT: Computer Science

PREVIOUS COURSES REQUESTED: Object-oriented programming

OBJECTIVES: Introducing Java programming language and J2SE platform technologies. Description of various advanced programming techniques and modalities of implementing them using Java. Introducing J2ME and presentation of fundamental notions concerning mobile devices programming. Presentation Sun J2SDK 1.6 development kit. Presentation of various integrated development editors (IDE) for Java (NetBeans, Eclipse).


DESCRIPTION OF SEMINARY / LABORATORY WORKS: Problems concerning each individual course.

TEACHING METHODS: Videoprojection


EVALUATION CONDITIONS:
- Each laboratory will contain two problems, each counted with 1 point.
- The exam will contain 20 questions, each counted with 1 point.
- Supplementary work may count additional points.

EVALUATION CRITERIA:
- To enter exam, each student must have 8 points.
- To pass exam, each student must have 5 points.

EVALUATION METHODS:
- Problem presentation (during semester) and Exam (in session)

FINAL RESULT - FORMULA:
- Gauss curve on the total number of points
- 5% = 10, 10% = 9, 20% = 8, 30% = 7, 25% = 6, 10% = 5
**COURSE NAME**: SOFTWARE ENGINEERING  
**CODE**: CS2209

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/OP-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>2</td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>

**COURSE TEACHER**: ASSIST. DR. ADRIAN IFTENE  
**DEPARTMENT**: Computer Science

**PREVIOUS COURSES REQUESTED**  

**OBJECTIVES**: Building a professional attitude towards software development. Students learn advanced methods and techniques enabling them to develop quality software artifacts while observing the customer's demands for functionality, costs and deadline.


**DESCRIPTION OF SEMINARY / LABORATORY WORKS**: The students must implement a project with a medium complexity, following the steps presented at courses: Requirements engineering. Building of UML diagrams. Using of Design patterns. Testing of the Project using manual and automatic testing. Project evaluation using Software metrics.

**TEACHING METHODS**: Course slide presentations. Lecture notes and tutorials available electronically.

**BIBLIOGRAPHY (SELECTION)**:  
- Craig Larman: *Applying UML and Patterns*, Addison Wesley, 2002  
- Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison Wesley, 1998

**EVALUATION**  
**conditions**  
Lab activity  
Project development  
Exam

**criteria**  
Minimum lab score 50% from maximum lab score (minimum 50% from labs tasks and minimum 50% from project score)  
Minimum exam score 40% from maximum exam score

**evaluation methods**  
Lab: weekly tasks, project  
Written exam – time for exam 30 minutes

**final result - formula**  
Final score is the sum of lab score, project score and exam score  
For the students who satisfy the evaluation criteria, the final grade is obtained by applying the Gauss curve on the final scores:  
grade 10 – first 5%  
grade 9 – next 10%  
grade 8 – next 20%  
grade 7 – next 30%  
grade 6 – next 25%  
grade 5 – last 10%  
The student who takes part in the exam receives a grade, otherwise he is considered absent. If any of the evaluation criteria are not met, the student receives a grade lesser than or equal to 4.
# DBMS PRACTICE

**Course Name:** DBMS PRACTICE  
**Code:** CS2210  
**Study Year:** II  
**Semester:** 2  
**Course Status:** (C-compulsory/OP-optional/F-facultative) C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C S L Pr.</td>
<td></td>
<td></td>
<td></td>
<td>(P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td>Romanian</td>
</tr>
<tr>
<td>1 - 2 - 42 -</td>
<td>94</td>
<td>5</td>
<td></td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. VICTOR FELEA</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

## Previous Courses Requested
- 

### Objectives
- Expression of complex queries using the language of SELECT commands.
- The Design of database Schemes for real applications.

### General Description
- The expression of complex queries using SELECT commands, the processing of trees using SELECT- SQL Oracle command, general languages for database queries, running plans for realizing of queries, design algorithms for database schemes.

### Description of Seminary / Laboratory Works
- Laboratory works: The expression of complex queries for some real queries. The realizing of a project that contains complex queries, the design of database schemes, indexes, using ORACLE management system for database.

### Teaching Methods
- Presentation of the course content, verification of individual projects realized in Oracle.

### Bibliography

2. Date C.J. Baze de date, traducere din engleză de Simona Preda și Titu Preda, Editura Plus, 2005.

### Evaluation
- **Conditions:** Laboratory Activity (Lab), Written Test (L)
- **Criteria:**
  - \( L \geq 20, \text{Lab} \geq 20 \)
  - \( \text{Max}[L]=50p, \text{Max}[\text{Lab}]=50p \)
- **Evaluation Methods:**
  - Lab in 7-th week.
  - Lab in laboratories.
- **Final Result - Formula:** The marks depend on the percentages fixed by the university.
COURSE NAME: LOGIC PROGRAMMING

COURSE CODE: CS2211O1

STUDY YEAR: II
SEMESTER: 2
COURSE STATUS (C- compulsory/ O- optional/ F- facultative): O

HOURS PER WEEK

C  S  L  Pr.  TOTAL HOURS PER SEMESTER  TOTAL HOURS INDIVIDUAL ACTIVITY  CREDITS  EVALUATION (P- during the semester, C- oral examination, E- written examination, M- mixed)  TEACHING LANGUAGE
2  -  2  -  56  64  4  M  Romanian

COURSE TEACHER
TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME: DRD. ALAIBA VASILE
DEPARTMENT: Computer Science

PREVIOUS COURSES REQUESTED: Logics for Computer Science

OBJECTIVES
1. Getting minimal skills for programming in “PROLOG-based” computer languages.
2. Learning an existing language implementation.

TÉMATICA GENERALĂ
1. First-order predicate logic and resolution in FOL (LP1)
2. Horn formulae and (definite) logic programs.
3. Refinements (restrictions and strategies) of resolution.
5. Non-determinism and negation.
6. The construction of a PROLOG-like interpreter.
7. Alternate Logic Programming paradigms

DESCRIPTION OF SEMINAR / LABORATORY WORKS
The themes will be fixed by the teacher at the beginning of the semester, according to the specified context. They will include the knowledge verification for the previous courses, short examples to be programmed and executed and larger projects.

TEACHING METHODS
All the classical didactic methods will be used: systematic exposure of knowledge, conversation, learning „by discovery”, etc. The Courses will be taught using a retro- or video-projector. A recent language implementation, SWI-PROLOG will be taught at the laboratory classes.

BIBLIOGRAPHY (SELECTION)

EVALUATION

conditions
Attendance is mandatory at the practical laboratories.

criteria
The on-line tests during the normal practical activity (laboratory) may generate 90 points (maximum, see below). To “graduate” the course, a minimum of 50 points is need.

evaluation methods
Concerning the “lab”. During the semester activity, there will be 5 themes (each quoted at 10 points, maximum) and a more complex project (quoted at 40 points, maximum).

final result - formula
The final grade is computed by first summing up all the obtained points and applying a Gauss-like distribution, according to existing regulations.
# Course: Functional Programming

**Course Name:** Functional Programming  
**Code:** CS2211O2

## Study Year and Semester
- **Study Year:** II  
- **Semester:** 2

## Course Status
- **COURSE STATUS:** (C- compulsory/OP- optional/ F- facultative)
  - **OP**

## Hours Per Week

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>Total Hours per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

## Course Teacher
- **TEACHER:** Prof. Dr. Gheorghe Grigoras  
- **DEPARTMENT:** Computer Science

## Previous Courses Requested
- **OBJECTIVES:** Understanding the concepts of functional programming, ability and skills in programming language Haskell. The concepts presented will include: functions, types, lists, abstract data types, high-order functions, side effect, lazy evaluation. Students will use language Haskell for implementation of data types and algorithms for solving problems.

## General Description
- **GENERAL DESCRIPTION:** Introduction to functional programming, The Hugs system, Introduction to the language Haskell, types and classes, polymorphism, recursive function, lists, trees, high-order functions, lazy evaluation, monad, definition of new types, Case Studies.

## Description of Seminar/Laboratory Works
- **DESCRIPTION OF SEMINARY / LABORATORY WORKS:** Hugs system, Programming in Haskell, definition of functions, recursive functions and high-order functions, interactive programs, 2 individual projects.

## Teaching Methods
- **TEACHING METHODS:** Slides with course items; seminar themes; projects’ issues; electronic version of the course; main readings will be find on the web page.

## Bibliography
- **BIBLIOGRAPHY (SELECTION):**
  - Limbajul Haskell: [www.haskell.com](http://www.haskell.com).
  - Ro/Haskell: [http://www.haskell.org/haskellwiki/Ro/Haskell](http://www.haskell.org/haskellwiki/Ro/Haskell)

## Evaluation
- **EVALUATION:**
  - **Conditions laborator activity (LA), final exam (FE)**
  - **Criteria:**
    - LA >= 5, FE >= 5
  - **Evaluation methods:** Mixed (during the semester and examination)
  - **Final result – formula:** Formula of the final score: 50% LA + 50% FE and ECTS criteria
OBJECTIVES
Students will learn to use a language adequate to the Computer Science field. Students will gain the necessary skills to participate in a job interview, will be able to write a CV and a covering letter in English, etc.

GENERAL DESCRIPTION OF THE COURSE
Grammar: “if” clauses, adjectives, sequence of tenses, prepositions. Vocabulary: specific to Computer Science + other fields of interest for the students. Topics of discussion: Artificial Intelligence, Cryptography, Windows vs Linux, Dependence on computers, etc.

DESCRIPTION OF SEMINARY / LABORATORY WORKS
The students will present a project in order to prove their ability of using accurate English and of expressing their ideas by means of a vocabulary adequate to the themes they present.

TEACHING METHODS
Interactive methods used all along the seminar.

PREVIOUS COURSES REQUESTED
-
# DESIGN AND ANALYSIS OF ALGORITHMS

**CODE:** CS3101

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/O-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>56</td>
<td>94</td>
</tr>
</tbody>
</table>

**COURSE NAME:**

**TEACHER:**

**DEPARTMENT:**

**PREVIOUS COURSES REQUESTED:** Algorithms and Programming (CS1101)

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
<th>GENERAL DESCRIPTION</th>
<th>DESCRIPTION OF SEMINARY / LABORATORY WORKS</th>
<th>TEACHING METHODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding algorithm analysis, algorithm design techniques, handling intractable problems.</td>
<td>Advanced data structures, design paradigms (greedy, dynamic programming, divide-et-impera, backtracking, branch-and-bound), NP-complete problems, modern heuristics.</td>
<td>Exercises helping to understand the algorithms, problem solving.</td>
<td>Slide-based intercative presentations, problem solving.</td>
</tr>
</tbody>
</table>

**BIBLIOGRAPHY** (SELECTION):


**EVALUATION**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Seminar activity (SA), home-works (HW), Final written test (WT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>SA &gt;= 5, LA &gt;= 5, FE &gt;= 5</td>
</tr>
<tr>
<td>Evaluation methods</td>
<td>Mixed</td>
</tr>
<tr>
<td>Final result – formula</td>
<td>10% AS + 40% HW + 50% WT + Bonus for extra activity</td>
</tr>
<tr>
<td>COURSE NAME</td>
<td>INFORMATION SECURITY</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>STUDY YEAR</td>
<td>III</td>
</tr>
<tr>
<td>SEMESTER</td>
<td>1</td>
</tr>
<tr>
<td>COURSE STATUS</td>
<td>(C-compulsory/OP-optional/F-facultative)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>56</td>
<td>94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>FACULTY/DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. FERUCIO LAURENTIU TIPLEA</td>
<td>Department of Computer Science</td>
<td></td>
</tr>
</tbody>
</table>

| PREVIOUS COURSES REQUESTED | No prerequisite required. |

| OBJECTIVES | This course is an advanced introduction to the theory of information and computer security. Students will gain basic and advanced knowledge about secure communications in the business world (e-commerce, e-payment, e-lotteries, e-gambling etc.) and how to secure their computers against electronic attacks. |

<table>
<thead>
<tr>
<th>GENERAL DESCRIPTION</th>
<th>The course includes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• brief introduction to cryptographic primitives (cryptosystem, digital signature, hash function, PKI);</td>
<td></td>
</tr>
<tr>
<td>• security protocols and policies, models of security;</td>
<td></td>
</tr>
<tr>
<td>• protocols for e-commerce, e-payment, identification and authentication, password management;</td>
<td></td>
</tr>
<tr>
<td>• smart-card technology;</td>
<td></td>
</tr>
<tr>
<td>• network security (IPsec, SSL, Kerberos, VPN, wireless, firewall);</td>
<td></td>
</tr>
<tr>
<td>• system security;</td>
<td></td>
</tr>
<tr>
<td>• e-mail security;</td>
<td></td>
</tr>
<tr>
<td>• security of mobile devices.</td>
<td></td>
</tr>
</tbody>
</table>

| DESCRIPTION OF SEMINARY / LABORATORY WORK | Seminars and laboratories are grouped around the chapter currently discussed in the course. They aim to illustrate the topics of the chapter mainly by practical applications. |

| TEACHING METHODS | On-line and blackboard presentation. |

|---------------------------|• PGP, Ipsec, SSL, Kerberos etc. documentation. |

<table>
<thead>
<tr>
<th>EVALUATION</th>
<th>conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>7 homeworks and a final exam</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>50% from the homeworks and 50% from the final exam</td>
</tr>
</tbody>
</table>
COURSE NAME: ARTIFICIAL INTELLIGENCE

CODE: CS3103

STUDY YEAR: III SEMESTER: 1 COURSE STATUS: (C-compulsory/OP-optional/F-facultative) C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td></td>
</tr>
<tr>
<td>C S L Pr.</td>
<td>2 2 -</td>
<td>56 94</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

COURSE TEACHER: PROF. DR. DAN CRISTEA

DEPARTMENT: Computer Science

PREVIOUS COURSES REQUESTED: -

OBJECTIVES: Introducing fundamental notions of the domain of Artificial Intelligence, beginning with the language Lisp, then presenting notions of knowledge representation, search in the state space, inferences in semantic networks and problems of planning and games.

GENERAL DESCRIPTION:

Chapter I Introduction: Definition of the domain of AI, the Turing test, philosophical problems, sub-domains of AI

Chapter II Lisp: Pure Lisp, s-expressions, syntax, representations of lists, evaluation of expressions, functions and macros, variables and their domains, closures, transfer of arguments in functions, recursivity

Chapter III Representation of knowledge and reasoning: Human-environment interactivity, descriptive semantic networks (queries, demons, the system IURES), event-oriented semantic networks

Chapter IV Production systems: Modelling AI problems, control, searching for solutions in state spaces, irrevocable strategies (hill-climbing), tentative strategies (backtracking hill-climbing), systematic search (depth-first search, breadth-first search, best-first), cost-guided search in graphs

Chapter V Planning and games: Search in game trees, the MIN-MAX method, the alpha-beta method, robot planning, STRIPS rules, bringing the robot back in the plan

DESCRIPTION OF SEMINARY / LABORATORY WORKS: Follows the themes taught at the course. Labs are posted at the address http://thor.info.uaic.ro/~orar/profesori/orar_ionita.html

TEACHING METHODS: Power Point presentations and interactive Lisp sessions


EVALUATION:

conditions: Minimal request to enter the exam: 26 lab points (out of 36) + 50 project points (out of 100) Minimum to pass: 26 lab points + 50 project points + 50 written exam points (out of 100)

criteria: Labs: 12*● in the class, ●● solved exercise, ●●● remarked) ➔ max 36; project: 0 – 100; written exam: 0 – 100

evaluation methods: Lab, project, written exam

final result - formula: (1,1*100/36*lab+1,2*pro+ex)/30, modified by Gauss
## APPLICATIONS DEVELOPMENT USING .NET FRAMEWORK

### CODE: CS3104

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/OP-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>III</td>
<td>1</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C S L Pr.</td>
<td></td>
<td></td>
<td></td>
<td>(P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td></td>
</tr>
<tr>
<td>2 - 2 - 56 5</td>
<td>94</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>INF. IOAN ASIMINOAEI</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
<th>Algorithms and Programming (CS1101)</th>
</tr>
</thead>
</table>

### OBJECTIVES

Teaching fundamental concepts in object-oriented programming and the use of the programming language C# on .NET Framework.

### GENERAL DESCRIPTION


### DESCRIPTION OF SEMINARY / LABORATORY WORKS


### TEACHING METHODS

Slides with course items; seminar themes; projects’ issues; electronic version of the course; main readings will be find on the web page.

### BIBLIOGRAPHY (SELECTION)

- Scott McLean, James Naftel, Kim Williams: Microsoft .NET Remoting. 2002
- Andrew Troelsen: Pro C# 2008 .NET 3.5 Platform
- Chris Sells, Michael Weinhardt: Windows Forms 2.0 Programming
- MSDN

### EVALUATION

<table>
<thead>
<tr>
<th>Conditions Participation to laborator hours (LA), participation to writing tests (WT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
</tr>
<tr>
<td>LA &gt;= 6, WT &gt;= 4</td>
</tr>
<tr>
<td>Final result – formula</td>
</tr>
</tbody>
</table>

Conditions Participation to laborator hours (LA), participation to writing tests (WT)
The main aim of the course is to provide grounding in stochastic processes and their application in various fields like economy, finance, biology, medicine, genetics, and physics. It is a useful tool for: (i) understanding the models of some phenomena which have a random development as well as the hypothesis on which these models are designed; (ii) developing models in C++/JAVA.

On completion of the course the students will be able to: (i) describe the principles of modeling; (ii) describe the general principles of stochastic processes and their classification into different types; (iii) define and apply a Markov chain and a Markov process; (iv) define and apply the main concepts of Gauss-Wiener processes and other Lévy processes; (v) explain and apply the concepts of "Monte Carlo" simulation of a stochastic process.

Science: Models of genetic population (i.e. Wright Model, Feller Model); Finance: introduction to basic financial concepts which will be used during course and seminars development: (i) types of contracts and ways they are used on financial market: working hypothesis of financial market, examples – types of options (European, American, Asian, Exotic, Bermudan, Russian, Parisan) which development are important challenges for mathematicians, computer scientist (hedgers, speculators, arbitrageurs, bulls and bears); (ii) Basic hypothesis for models development (i.e. arbitrage free); (iii) Binomial Models: - Modele utilizate la bursă (Wall Street): Cox-Ross-Rubinstein (CRR) Model, Hull and White Model, Cox-Ingersoll-Ross Model, Brace-Gatarek-Musiela Model, Change of numeraire, Merton formula, Simulation techniques: Monte Carlo Method (describe how apparently pseudo-random integers can be generated using a computer; describe how pseudo-random drawings from specified distributions can be generated; explain how a series of sets of correlated normal random variates can be generated; explain the disadvantages of using random as opposed to pseudo-random numbers; discuss how to decide how many simulations to carry out for any particular purpose); Implementation of other programming techniques: (JAVA/C++) for some of the most important models used on the Wall Street (Hull-White, CIR, HJM, etc.)

Evaluations: Mixed (during the semester and examination)
Criteria: Seminar work (five sets of problems to work for understanding), a project (prepared by a team composed of max 2 students and supervised by the professor in charge)

Evaluation methods: Mixed (during the semester and examination)
Final result – formula: 70% evaluation during semester (30% seminar work + 40% for the understanding of the issues associated with the project), 30% final exam (project evaluation)

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C  S  L  Pr.</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY (SELECTION)

OBJECTIVES
This course is an advanced introduction to theoretical computer science emphasizing two interrelated areas: the theory of computability (how to tell whether problems are algorithmically solvable) and the theory of complexity (how efficient an algorithm is).

GENERAL DESCRIPTION
The heart of the course material on computability is the recursion theory (primitive recursion and mu-recursion). The connection with Turing machines and WHILE-programs is made, and some important decision problems are discussed. The complexity theory is considered both from qualitative and quantitative points of view: dynamic complexity measures are introduced, the concept of a complexity class as a class of languages is presented, the strict correspondence between such classes and decision problems is established, and techniques used to study properties of such classes are formulated. Then, we study in details the basic properties of some of the most significant complexity classes. Finally, we deal with probabilistic algorithms and with the corresponding complexity classes.

DESCRIPTION OF SEMINARY/LABORATORY WORK
Seminars and laboratories are grouped around the chapter currently discussed in the course. They are aimed to illustrate the topics of the chapter mainly by practical applications.

TEACHING METHODS
On-line and blackboard presentation.

BIBLIOGRAPHY (SELECTION)
- Journal papers.

EVALUATION
conditions
criteria
evaluation methods 6 homeworks and a final exam
final result - formula 50% from the homeworks and 50% from the final exam
COURSE NAME: NUMERICAL CALCULUS

CODE: CS3207

STUDY YEAR: III  SEMESTER: 2  COURSE STATUS (C-compulsory/OP-optional/F-facultative): C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C  S  L  Pr.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Romanian</td>
</tr>
<tr>
<td>2   -    2    -</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>

COURSE TEACHER
LECT. DR. ANCA IGNAT

DEPARTMENT
Computer Science

PREVIOUS COURSES REQUESTED
Mathematics, Algorithms and Programming, Object-Oriented Programming

OBJECTIVES
Learning about numerical methods for approximation of continuous mathematical problems.

GENERAL DESCRIPTION
- Examples, floating point computing, types of errors, propagation of errors
- LU decompositions (Gauss elimination algorithm, Cholesky factorisation), QR decomposition (Givens and Householder algorithms), singular value decomposition
- Iterative methods for solving linear systems (Jacobi and Gauss-Seidel methods, successive overrelaxation)
- Eigenvalues and eigenvectors approximation (Jacobi method for symmetric matrices, QR type algorithms)
- Solving nonlinear equations and systems of nonlinear equations (Newton type methods, false position method, secant method, methods for the roots of polynomials)
- Polynomial interpolation (Lagrange polynomial, Newton polynomials), spline interpolation (linear continuous, cubic of class $C^2$)
- Numerical integration (Newton-Cotes type formulae)

DESCRIPTION OF SEMINARY / LABORATORY WORKS
- Evaluation of elementary functions (sin/cos/...), errors in numerical computations;
- Solving linear systems:
  1. Substitution method, LU decomposition;
  2. QR decomposition: Givens or Householder algorithm;
  3. Iterative methods: Jacobi and Gauss-Seidel methods;
- Jacobi method for finding the eigenvalues and eigenvectors for symmetric matrices;
- Solving nonlinear equations: bisection method, Newton-Raphson method, false position method, secant method, methods for approximating roots of polynomials;
- Polynomial interpolation: Newton-Lagrange polynomial, Aitken algorithm, $C^2$-cubic spline functions;
- Numerical integration: Newton-Cotes type formulae, iterate methods.

TEACHING METHODS
Course – using videoprojector, Laboratory works - files describing the algorithms

BIBLIOGRAPHY (SELECTION)
- C. Ignat, C. Ilioi, T. Jucan, Elemente de informatică și calcul numeric, Editura Univ. „Al.I. Cuza” Iași, 1989,
- T.A. Beu, Calcul numeric în C, Editura Albastră, Cluj, 2000,
- V. Iorga, B. Jora, Metode numerice, Ed. Albastra, Cluj, 2004

EVALUATION
- Evaluation conditions
  - Implementing the homeworks
  - Final score (lab + exam) must exceed a certain threshold
- Evaluation criteria
  - Each homework (there are at most 8 homeworks) has a maximal score associated if the deadline is respected (the homeworks can be presented after the deadline but with penalties)
  - The exam consists of exercises with all resources available (evaluation between 1 - 10)
- Evaluation methods
  • Written exam
  • Evaluation of the implemented homeworks

final result - formula
according to the present regulations applied to the final score (laboratory score + exam score)
# Computer Graphics

**Course Name:** Computer Graphics  
**Code:** CS3208  
**Study Year:** III  
**Semester:** 2  
**Course Status:** C (compulsory)

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P: during the semester, C: oral examination, E: written examination, M: mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C  S  L  Pr.</td>
<td>- 2 - 2 - 56</td>
<td>94</td>
<td>5</td>
<td>E</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

**Previous Courses Requested:** Algorithms and Programming, Object Oriented Programming

**Objectives:**
- Introduction to computer graphics.
- Acquiring the ability of designing simple models (i.e., a collection of statically or dynamically objects having simple geometric shapes).
- Acquiring the techniques of rendering models (by using raster graphics).
- Acquiring the ability of designing graphics software using a standard graphics API.

**General Description:**
1. Introduction to Computer Graphics.
3. 2D/3D Geometric Transformations and their representation by matrices.
5. 3D Viewing Transformations.
7. Spatial Subdivision Techniques.
10. Textures.
12. OPENGL Library.

**Description of Seminary/Laboratory Works:**
2. Color.
3. Geometric Transformations in OPENGL.
4. 2D Raster Graphics.
5. 3D Viewing Transformations in OPENGL.
6. Reflection and Shading Models, Textures in OPENGL.

**Teaching Methods:** Lectures are given using MS Office PowerPoint and blackboard presentations. During the laboratory classes the students will receive a set of solved problems (usually programming assignments) and they are asked to solve a set of similar problems or to implement (using OpenGL and Visual Studio 6.0) some algorithms schematically presented during the lectures.

**Bibliography (Selection):**

**Evaluation:**
- **Conditions:** No absence at laboratory classes. In order to pass the exam the students are required to solve completely at least 25% of homeworks and to solve completely at least 25% of the written test.
- **Criteria:** Activity at laboratory classes (pertinent questions regarding the assignments, presentation of several stages of solving the assignment), strong lectures attendance (in case of low levels of attendance, the attendance of a particular student could be recompensed by granting bonuses to his/her final grade).
- **Evaluation Methods:** Programming assignments at laboratory, written test in the last week of the semester or in the exam session.
- **Final Result - Formula:** A partial grade is established based on the lab results and the written test results. Then the students are classified based on their partial grade, the final grade being assigned accordingly to ECTS grading system—European Credit Transfer System and Diploma Supplement.
OBJECTIVES
The course is an introduction to a programming paradigm that is suitable for creating expert systems. An expert system (ES) is a program that simulates the experience and the reasoning capacities of a human specialist in a certain field.

GENERAL DESCRIPTION
Internal ES design, types of ESs (forward chaining, backward chaining, and mixed chaining), the RETE algorithm (the foundation for optimizing the ES implementations). CLIPS and PERL languages are introduced. Programming techniques in CLIPS are presented in an interactive manner, accompanied by numerous examples.

DESCRIPTION OF SEMINARY/LABORATORY WORKS
The laboratories are used for practicing the writing of programs in CLIPS and PERL. Part of the time will be allocated to working on a project.

TEACHING METHODS
Power Point presentations and interactive programming sessions

BIBLIOGRAPHY
Dan Cristea, "Programarea bazată pe reguli", Ed. Academiei, 2002

EVALUATION
conditions
Minimal request to enter the exam: 30 lab points (out of 42) + 20 project points (out of 100) or 20 points on the partial exam
Minimum to pass: 26 lab points + 50 project points + 50 written exam points (out of 100)

criteria
Labs: 12*(● in the class, ●● solved exercise, ●●● remarked) → max 36; project: 0 – 40; written exam (partial + final): 0 – 40

evaluation methods
Lab, project, written exam

final result - formula
(lab+pro+ex)/10, modified by Gauss
**COURSE NAME** BIOINFORMATICS

**STUDY YEAR** III **SEMESTER** 2 **COURSE STATUS** (C- compulsory/ O- optional/ F- facultative)

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 C</td>
<td>2 S</td>
<td>2 L</td>
<td>Pr.</td>
<td>48</td>
<td>144</td>
</tr>
</tbody>
</table>

**TEACHING AND SCIENTIFIC DEGREE**

**TEACHERS**

ASSOC. PROF. DR. LIVIU CIORTUZ

**DEPARTMENT**

Computer Science

**PREVIOUS COURSES REQUESTED**

- 

**OBJECTIVES**

Understanding the basic technics in the analysis of genetic sequences

**GENERAL DESCRIPTION**

1. Fundamental notions in molecular biology
2. Hidden Markov Models (HMMs)
3. Alignment of pairs of DNA/protein sequences
4. Alignment of pairs of DNA/protein sequences using pair-HMMs
5. Multiple alignment of DNA/protein sequences
6. Multiple alignment of DNA/protein sequences using profile-HMMs
7. Phlogenetics; probabilistic models
8. Probabilistic context free grammars (PCFGs)
9. Analysis of RNA sequences using PCFGs
10. Motif in identification in genetic sequences

**DESCRIPTION OF SEMINARY/ LABORATORY WORKS**

Implementation of some of the presented algorithms and their applications. Presentation of recent papers in the area of bioinformatics.

**TEACHING METHODS**

Slides on video-projector

**BIBLIOGRAPHY (SELECTION)**

"Biological Sequence Analysis", Durbin, Eddy, Krogh, Mitchison; Cambridge University Press, 1998


**EVALUATION**

- **conditions**
  - Minimum 1.5 (of 4) points for lab, minimum 1.5 (of 4) points at the final exam

- **evaluation methods**
  - Basis (2 points) + lab work (4 points) + final exam (4 points)
**THE MODELLING OF DISTRIBUTED SYSTEMS USING PETRI NETS**

**STUDY YEAR** III  |  **SEMESTER** 2  |  **COURSE STATUS** (C-Compulsory/OP-Optional/F-Facultative)  |  OP

**HOURS PER WEEK**

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS OF INDIVIDUAL ACTIVITY</th>
<th>CREDIT S</th>
<th>EVALUATION TYPE</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>P</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

**TAUGHT BY**

<table>
<thead>
<tr>
<th>ACADEMIC AND SCIENTIFIC TITLE, NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIST. DRD. OANA OTILIA PRISECARU</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

**REQUIRED COURSES**

- 1. Assimilating the basic notions about Petri nets and their different extensions.
- 2. Acquiring analysis techniques for Petri nets.
- 3. Understanding the practical applications of Petri nets in various areas.
- 4. Achieving the ability to model and analyze real systems by using different types of Petri nets.

**OBJECTIVES**

**GENERAL THEMATICS**

2. Applications of P/T nets (modeling communication protocols, distributed algorithms).
5. Applications of coloured Petri nets in industry.
6. Timed Petri nets.
7. Applications of timed Petri nets.

**SEMINARY / LABORATORY THEMATICS**


**TEACHING METHODS**

- Exposure using course notes (slides, available from the beginning of each course) presented with a video-projector, demos on computer.
- Exposure (video-projector for the course), debate, exercises, problems, case studies (laboratory).

**BIBLIOGRAPHY**


**EVALUATION**

<table>
<thead>
<tr>
<th>conditions</th>
<th>At least 20 points for the seminary and laboratory activity (LSA). At least 20 point at the written test (TS).</th>
</tr>
</thead>
<tbody>
<tr>
<td>criterias</td>
<td>A maximum of 100 points can be accumulated.</td>
</tr>
<tr>
<td>modes</td>
<td>Seminary and laboratory activity (exercises, report): 50 points. Written test (TS) in the 14th week: 50 points.</td>
</tr>
<tr>
<td>formula</td>
<td>The final mark is computed by summing the scores (LSA + TS) and then applying Gauss distribution.</td>
</tr>
</tbody>
</table>
OBJECTIVES
This course offers a thoroughgoing study of the basic mechanisms of operating systems and focuses on advanced topics about the kernel modules of operating systems. The operating systems concepts are explained using the Windows XP and Windows Server 2003 operating system family, based on the curriculum developed by the Microsoft Windows Academic Program, structured according with ACM/IEEE Operating System Body of Knowledge.

GENERAL THEMATICS
1. Overview of operating systems
2. Review of basic concepts about operating systems design, concurrency, scheduling, memory management
3. Device management. I/O system. File systems
4. Real-time and embedded systems
5. Fault tolerance
6. System performance evaluation and troubleshooting
7. Scripting
8. Comparing the Linux and Windows kernels
9. Windows and Unix interoperability

SEMINARY / LABORATORY THEMATICS
Lab works related to the course' general thematics. The students will do lab experiments and assignments through which they will study operating systems concepts related to Windows XP and Windows Server 2003 operating system family, using the resources from Windows Academic Program.

TEACHING METHODS
Exposure using course notes (slides, available from the beginning of each course) presented with a video-projector, demos on computer.

BIBLIOGRAPHY

EVALUATION
conditions The presence at the laboratory activities and at the written thesis.
criteria Minimal score for graduation is 30p
modes Evaluation during the semester (lab works) and a final written thesis.
formula Final Score = Lab * 0.5 + WrittenThesis * 0.5.
The final mark is obtained from the final score through classification based on the ECTS – European Credit Transfer System and Diploma Supplement.
<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>94</td>
</tr>
</tbody>
</table>

**COURSE NAME**: EMBEDDED SYSTEMS

**CODE**: CS31102

**STUDY YEAR**: III  
**SEMESTER**: 2  
**COURSE STATUS**: (C-compulsory/OP-optional/F-facultative)  
**OP**: OP

**HOURS PER WEEK**

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>Total Hours Per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>94</td>
<td>5</td>
<td>M</td>
</tr>
</tbody>
</table>

**TEACHING LANGUAGE**: Romanian

**TEACHING AND SCIENTIFIC DEGREE**, **FIRST NAME**, **LAST NAME**: LECT. DR. VLAD RĂDULESCU

**DEPARTMENT**: Computer Science

**PREVIOUS COURSES REQUESTED**: Computer Architecture and Operating Systems, Hardware Practice, Operating Systems

**OBJECTIVES**

- Getting acquainted with the concept of embedded systems.
- Learning the hardware design requirements.
- Learning the ways of writing software for embedded systems.

**GENERAL DESCRIPTION**

- Hardware and software design of embedded systems.
- Simple and complex hardware automata. Hardware description languages.
- Real-time systems. Restrictions.

**DESCRIPTION OF SEMINARY / LABORATORY WORKS**

- Simple and complex hardware automata.
- The Verilog hardware description language.
- Buses. Arbitration techniques.
- Programming the microcontrollers.

**TEACHING METHODS**

- Exposition, problem-solving, case studies, exercises.

**BIBLIOGRAPHY**


**EVALUATION**

- **conditions**: The presence at the laboratory activities.
- **criteria**: At least 5 points on each test; 3 or 4 points are accepted for at most one test.
- **evaluation methods**: Two written tests, the first (T1) during the 7th week, the second (T2) during the examination session.
- **final result - formula**: Final result = (T1 + T2) / 2, on which Gaussian distribution is applied.
# ADVANCED SOFTWARE ENGINEERING TECHNIQUES

**COURSE NAME:** ADVANCED SOFTWARE ENGINEERING TECHNIQUES  

**CODE:** MCG1101

**STUDY YEAR:** 1  
**SEMESTER:** 1  
**COURSE STATUS:** (C-compulsory/OP-optional/F-facultative) C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C 2</td>
<td>S 2</td>
<td>L 2</td>
<td>Pr. 2</td>
<td>56</td>
<td>124</td>
</tr>
</tbody>
</table>

**TEACHING LANGUAGE:** Romanian

**TEACHER:** PROF. DR. BAZIL PÂRV  
**DEPARTMENT:** Computer Science

**PREVIOUS COURSES REQUESTED**

**OBJECTIVES**  
The development of a good understanding of the software engineering field. Students will get to learn concepts used in designing complex systems, modern architectures and new approaches in modelling competitive software.

**GENERAL DESCRIPTION**

0 SWEBOK: THE ROLE OF SOFTWARE ENGINEERING  
WWW.SWEBOK.ORG

KNOWLEDGE AREAS

RELATED COURSES

STEVE MCCONNELL (WWW.CONSTRUX.COM)

PRAGMATIC PROGRAMMER

1 OO DESIGN
- CLASSES: GRASP (LARMAN), RESPONSIBILITY-DRIVEN DESIGN
- MID-LEVEL: GOF (OVERVIEW)
- HIGH-LEVEL: ARCHITECTURAL STYLES (PATTERNS), SOA
- OO DESIGN PRINCIPLES

2 SYSTEM DEVELOPMENT AND MAINTENANCE

THE CHARACTERISTICS OF A GOOD ARCHITECTURE (RCM)

AGILE MODEL-DRIVEN DEVELOPMENT

ENTERPRISE APPLICATION ARCHITECTURE PATTERNS

DOMAIN-DRIVEN DESIGN: CONCEPT AND PATTERNS

TEST-DRIVEN DEVELOPMENT

REFACTORING: CODE (R IN THE SMALL), ARCHITECTURE (R IN THE LARGE)

3 MODELING

BUSINESS MODELING: BPMN

UML BEHAVIORAL MODELS: STATE MACHINES, ACTIVITIES

WORKFLOW PATTERNS

MODEL-DRIVEN DEVELOPMENT, MODEL-DRIVEN ARCHITECTURE

DOMAIN SPECIFIC LANGUAGES (DSL), FRAMEWORKS: ECLIPSE MODELING FRAMEWORK, OPEN ARCHITECTURE WARE (OAW)

**DESCRIPTION OF SEMINARY / LABORATORY WORKS**

Applying software engineering best practices.

Refactoring, improving the design of existing code.

Automated testing

Advanced design patterns

**TEACHING METHODS**

Course slide presentations. Lecture notes and tutorials available electronically.

**BIBLIOGRAPHY**

(SELECTION)


The Pragmatic Programmer: From Journeyman to Master by Andrew Hunt and David Thomas


**EVALUATION**

<table>
<thead>
<tr>
<th>conditions</th>
<th>Participation in practical works. Accumulation of 50 points, according to the final grade formula.</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>Written exam (T, max. 40), article (R, max. 50), practical project (P, max. 60)</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>Written exam at the end of the semester.</td>
</tr>
<tr>
<td>The article and the project will be evaluated as part of the laboratory works.</td>
<td></td>
</tr>
<tr>
<td>final result - formula</td>
<td>On the final grade a Gauss like distribution will be applied, according to the current regulations.</td>
</tr>
</tbody>
</table>
# OPERATIONS RESEARCH

<table>
<thead>
<tr>
<th>COURSE NAME</th>
<th>OPERATIONS RESEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE:</td>
<td>MCG1102</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>I</th>
<th>SEMESTER</th>
<th>I</th>
<th>COURSE STATUS (C-compulsory/O-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOURS PER WEEK</td>
<td>TOTAL HOURS PER ACTIVITY</td>
<td>CREDITS</td>
<td>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td>TEACHING LANGUAGE</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>SEMESTER</td>
<td>ACTIVITY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHERS</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSOC.PROF. DR. MARIANA RODICA BRÂNZEI</td>
<td>Computer Science</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
<th>Mathematics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Probability Theory and Statistics</td>
</tr>
</tbody>
</table>

## OBJECTIVES
Operations Research (OR) is the application of scientific techniques and methodology to decision-making problems. The purpose of the course is to provide a sound understanding of several quantitative models and methods which are frequently and successfully applied in managerial decision making, and more generally, to demonstrate the possibilities and limitations of the quantitative approach in the analysis of decision situations.

## GENERAL DESCRIPTION
- Overview of the OR Modelling Approach;
- Linear Programming (LP); Graphical LP solutions; The Simplex Method;
- Duality Theory and Sensitivity Analysis;
- Integer Programming;
- Game Theory and Decision Analysis,
- Inventory Theory.

## DESCRIPTION OF SEMINARY / LABORATORY WORKS
The OR laboratory is application oriented and is mainly build on the case method. The case method creates a classroom in which students succeed to understand theory and exercise the skills of leadership and team work in the face of real problems. Several types of managerial decision problems are described which can be adequately represented by quantitative models and can be solved by mathematical methods. The emphasis is on model formulation. Computer solutions by using spreadsheet programs (like Excel) and computer interpretations in the context of the decision situations are also demonstrated.

## TEACHING METHODS
Lectures using overhead projector and blackboard.

## BIBLIOGRAPHY (SELECTION)

## EVALUATION
| conditions | ME (meadterm evaluation), FE (final evaluation), W (Laboratory works) |
| criteria | ME ≥ 6, FE ≥ 4, W ∈ {0, 1, 2} |
| evaluation methods | ME (written test (90 minutes) covers 1-6 weeks; FE (written test (90 minutes) covers weeks 8-13) |
| final result - formula | ME + FE + W |
COURSE NAME: JAVA TECHNOLOGIES

CODE: MCG1103

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>154</td>
<td>8</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

STUDY YEAR: I
SEMESTER: 1
COURSE STATUS (C- compulsory, O- optional, F- facultative): C

HOURS PER WEEK

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>154</td>
<td>8</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

STUDY YEAR: I
SEMESTER: 1
COURSE STATUS (C- compulsory, O- optional, F- facultative): C

HOURS PER WEEK

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>154</td>
<td>8</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

COURSES REQUESTED

Object-oriented programming, Advanced Techniques of Programming, Web Technologies, Networking

OBJECTIVES

Java Enterprise Edition (Java EE), Server-Side Programming

GENERAL DESCRIPTION

Introduction to Java Enterprise Edition (Java EE)
Web Components: Servlets, Filters, Java Server Pages (JSP), Custom Tag Libraries (CTL)
Template Engines: Velocity, Free Marker
MVC frameworks: Struts, Java Server Faces
Java Naming and Directory Interface (JNDI), Java Message Service (JMS).
Object-Relational Mappings: Hibernate, Java Persistence API (JPA)
Object-Oriented Databases DB4O. Aspect Oriented Programming (AOP): AspectJ.
Service Oriented Architectures (SOA), Enterprise Java Beans (EJB)

DESCRIPTION OF SEMINARY / LABORATORY WORKS

Problems concerning each individual course.

TEACHING METHODS

Videoprojection

BIBLIOGRAPHY (SELECTION)

Cristian Frasinaru, Curs practic de Java, Matrix Rom Bucuresti (2005), ISBN 973-685-856-1
Jayson Falkner, Kevin Jones, Servlets and Java Server Pages,
Ed Roman, Scott Ambler, Tyler Jewel, Mastering Enterprise JavaBeans
http://java.sun.com/javaee

EVALUATION

conditions
Each laboratory will contain two problems, each counted with 1 point.
The exam will contain 20 questions, each counted with 1 point.
Supplementary work may count additional points.
criteria
To enter exam, each student must have 8 points.
To pass exam, each student must have 5 points.
evaluation methods
Problem presentation (during semester) and Exam (in session)
final result - formula
Gauss curve on the total number of points
5%=10, 10%=9, 20%=8, 30%=7, 25%=6, 10%=5
# Advanced Artificial Intelligence (Machine Learning)

<table>
<thead>
<tr>
<th>COURSE NAME</th>
<th>ADVANCED ARTIFICIAL INTELLIGENCE (MACHINE LEARNING)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>MCG1205</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>1</th>
<th>SEMESTER</th>
<th>2</th>
<th>COURSE STATUS (C-compulsory/O-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C  S  L  Pr.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Romanian</td>
</tr>
<tr>
<td>2   -   2   -</td>
<td>56</td>
<td>184</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHERS</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assoc. Prof. Dr. Liviu Ciortuz</td>
<td>Computer Science</td>
<td></td>
</tr>
</tbody>
</table>

| PREVIOUS COURSES REQUESTED | - |

## Objectives

Understanding the conception and practical usage of the most important algorithms for classification and clustering

## General Description

1. Concept Learning and General-to-Specific Ordering  
2. Decision Trees  
3. Instance-based Learning  
4. Retele neuronale  
5. Bayesian Learning  
6. Support Vector Machines  
7. Clustering Algorithms  
8. Boosting, bagging, random forests, voting, co-training  
9. Evaluating Hypotheses  
10. Computational Learning Theory  
11. Reinforcement Learning

## Description of Seminar/Laboratory Works

Presentation of recent papers in the area of machine learning:  
- String kernels and applications  
- Pattern matching on strings  
- Grammar learning  

## Teaching Methods

Slides on video-projector

## Bibliography (Selection)

- “Data Mining: Practical ML Tools and Techniques with Java Implementations”, Witten, Frank; Morgan Kaufmann Publishers, 2000  
- “The Elements of Statistical Learning”, Friedman, Hastie, Tibshirani, 2001  
- “Pattern Matching and Machine Learning”, Ch. Bishop, 2006

## Evaluation

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Minimum 1.5 (of 4) points for lab, minimum 1.5 (of 4) points at the final exam</th>
</tr>
</thead>
</table>

| Evaluation Methods | Basis (2 points) + lab work (4 points) + final exam (4 points) |

---

55
COURSE NAME: WEB APPLICATION DEVELOPMENT
CODE: MCG2101

STUDY YEAR: II  SEMESTER: 1  COURSE STATUS (C-compulsory/O-optional/F-facultative) C

HOURS PER WEEK

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>56</td>
<td>124</td>
<td>8</td>
</tr>
</tbody>
</table>

COURSE TEACHER

TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME
ASSOC. PROF. DR. SABIN-CORNELIU BURAGA
DEPARTMENT
Computer Science

PREVIOUS COURSES REQUESTED
Web Technologies, Software Engineering

OBJECTIVES
Giving a general view regarding knowledge modeling in the context of evolution towards semantic Web. Students will achieve understanding about the development of knowledge management Web applications on the basis of the present Web technologies: metadata, microformats, taxonomies, thesauri, and ontologies.

GENERAL DESCRIPTION

DESCRIPTION OF SEMINARY / LABORATORY WORKS

TEACHING METHODS
Interactive presentations. Direct interaction. Online access to educational resources via the Website course.

BIBLIOGRAPHY (SELECTION)

EVALUATION

<table>
<thead>
<tr>
<th>conditions</th>
<th>1 project (P), 1 test during semester (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>project P&gt;5, test T&gt;5</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>1 project (P), 1 test during semester (T)</td>
</tr>
<tr>
<td>final result - formula</td>
<td>0.5 * P + 0.4 * T + 1</td>
</tr>
</tbody>
</table>
## Course Details

**Course Name:** Project Management  
**Code:** MCG2205

<table>
<thead>
<tr>
<th>Study Year</th>
<th>Semester</th>
<th>Course Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>2</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

### Hours per Week

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>L</th>
<th>Pr.</th>
<th>Total Hours Per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>56</td>
<td>184</td>
<td>8</td>
<td>M</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

### Course Teacher

**Name:** Prof. Dr. Gabriela Meșnîță  
**Department:** Faculty of Economics and Business Management

### Previous Courses Requested

Management; Software Engineering

### Objectives

- To create the project management feel
- To develop a specific language
- To get abilities for project design
- To achieve different information and knowledge concerning the project management, assessment and evaluation
- To use the project management methods and tools

### General Description

- Conceptual framework of project management
- Team work and team management of project management
- Practical approach of project plan
- Cost control and project budget
- Proposal project evaluation
- Project monitoring and control
- Project management and information technology

### Description of Seminary/Laboratory Works

Team work focused on design a project following: identification the right phases of project development; reference terms and requirements of different types of project; project initiation; project planning; Project assessment; use specific software (Microsoft Project).

### Teaching Methods

Interactive courses, practical home work, team work

### Bibliography (Selection)


### Evaluation

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Lab project; Home work; Partial evaluation at cours and lab; Writing exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>Each condition must be evaluated at least with 5</td>
</tr>
<tr>
<td>Evaluation methods</td>
<td>Project presentation (40%); Home work (10%); Partial evaluation (10%); Writing exam (40%)</td>
</tr>
<tr>
<td>Final result - formula</td>
<td>Project x 0.4 + Home work x 0.1 + Partial evaluation x 0.1 + Exam x 0.4</td>
</tr>
</tbody>
</table>
## COURSE NAME
PARALLEL ALGORITHMS AND PARALLEL PROGRAMMING

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/O-optiona/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>184</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LECT. DR. LUCIAN GHIRVU</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
<th>BSc in Computer Science</th>
</tr>
</thead>
</table>

### OBJECTIVES

- **PARALLEL ALGORITHMS**: this course is an introduction in the design and the analysis of parallel algorithms and also presents the main models of parallel computation.
- **PARALLEL PROGRAMMING**: there are presented its main concepts and applications (using the libraries such as Pthreads, MPI, OpenMP).

### GENERAL DESCRIPTION

- **PARALLEL ALGORITHMS**: parallel computer/algorithm, the analysis of parallel algorithms, parallel computation models (PRAM, interconnection networks, combinational circuits), parallel prefix computation, using the „Divide and conquer” paradigm in the design of parallel algorithms (for searching, merging, and selection), using of pointer-based data structures in parallel algorithms.
- **PARALLEL PROGRAMMING**: using of the libraries such as Pthreads, MPI, OpenMP.

### DESCRIPTION OF SEMINARY/LABORATORY WORKS

- Problems (projects).
- The implementation of some parallel algorithms and solving problems by using the libraries C Pthreads, MPI, and OpenMP.
- Case studies.

### TEACHING METHODS

- Exposition (lectures are given using blackboard presentations).

### BIBLIOGRAPHY (SELECTION)

- Papers from journals and conferences.

### EVALUATION

<table>
<thead>
<tr>
<th>Conditions</th>
<th>No absence at laboratory classes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>Activity at laboratory classes (pertinent questions regarding the assignments, presentation of several stages of solving the assignment), strong lectures attendance (in case of low levels of attendance, the attendance of a particular student could be recompensed by granting bonuses to his/her final grade).</td>
</tr>
<tr>
<td>Evaluation methods</td>
<td>Programming assignments at laboratory, written test in the last week of the semester or in the exam session, scientific paper presentations.</td>
</tr>
<tr>
<td>Final result - formula</td>
<td>A partial grade is established based on the results of the above evaluation methods. Then the students are classified based on their partial grade, the final grade being assigned accordingly to ECTS grading system—European Credit Transfer System and Diploma Supplement.</td>
</tr>
<tr>
<td>COURSE NAME</td>
<td>SECURITY PROTOCOLS</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>STUDY YEAR</td>
<td>1</td>
</tr>
<tr>
<td>SEMESTER</td>
<td>2</td>
</tr>
<tr>
<td>COURSE STATUS</td>
<td>(C- compulsory/ O- optional/ F- facultative)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR. CONSTANTIN ENEA</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

| PREVIOUS COURSES REQUESTED | |
|-----------------------------| |
| Familiarity with basic cryptographic concepts as treated, for example, in the core course "Information Security", is helpful, but can in principle also be acquired in parallel to attending the course. |

| OBJECTIVES | |
|------------| |
| The objectives of the course are to provide students with the practice of cryptographic protocols (efficient implementations, vulnerabilities, etc.) and to provide a background against which the student will be able to develop new protocols and applications. |

| GENERAL DESCRIPTION | |
|---------------------| |
| Topics include:  |
| • basics on security (cryptographic) protocols  |
| • formalisms: MSR, strand spaces  |
| • undecidability of secrets  |
| • bounded protocols  |
| • tagged protocols  |
| • recursive protocols  |
| • verification techniques: BAN, inductive method, strand spaces, based on I/O-automata |

| DESCRIPTION OF SEMINARY / LABORATORY WORK | |
|------------------------------------------| |
| All seminars will be oriented on the topic discussed during the courses. Students will be asked to prepare a research project. |

| TEACHING METHODS | |
|-------------------| |
| On-line and blackboard presentation. |

| BIBLIOGRAPHY (SELECTION) | |
|---------------------------| |
| • F.L. Tiplea: Introduction to Cryptography (in preparation) - chapters of the book will be available to students.  |
| • Research articles. |

| EVALUATION | |
|-------------| |
| conditions | |
| criteria | |
| evaluation methods | One project and a final exam. |
| final result - formula | 50% from the project and 50% from the final exam. |
**COURSE NAME** | COMBINATORIAL OPTIMIZATION | MOC1206
---|---|---
**STUDY YEAR** | 1 | **SEMESTER** | 2 | **COURSE STATUS** (C-compulsory/OP-optional/F-facultative) | C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>-</td>
<td>56</td>
</tr>
</tbody>
</table>

**COURSE TEACHER**

<table>
<thead>
<tr>
<th>PREVIOUS COURSES REQUESTED</th>
<th>Design and Analysis of Algorithms, Graph Algorithms, Operational Research</th>
</tr>
</thead>
</table>

**OBJECTIVES**

This course is an advanced study in the design and analysis of algorithms related to combinatorial optimization using SAT, a classical problem in theoretical computer science, which was widely investigated and has brought forward a rich body of methods and tools, both in theory and practice. To provide basis for independent research on the subject.

**GENERAL DESCRIPTION**

The course concentrates on the combinatorial and algorithmic aspects of the propositional logic Satisfiability Problem. The fundamental challenge is understanding of Combinatorial Search Spaces.

**DESCRIPTION OF SEMINARY / LABORATORY WORKS**

Each seminar debates recent research papers in order to deepen the subjects introduced in the course. All these papers are posted at the beginning of the semester such that interested students could try to study in advance.

**TEACHING METHODS**

Video presentations of the slides (containing the course notes) available in pdf format at the beginning of each class.

**BIBLIOGRAPHY**

- A repository of about 200 papers covering the material of the course will be distributed.
- Emo Welzl: Course on Satisfiability of Boolean Formulas – Combinatorics and Algorithms (http://www.ti.inf.ethz.ch/ew/courses/SAT08/)
- http://www.satlive.org/SATBIB/

**EVALUATION**

A student will be considered to have passed the exam if (s)he obtains at least 50 points.

- **Seminar activity (attendance, work quality):** 0-20 points.
- **Homeworks:** 0-40 points.
- **Written Final test:** 0-40 points.

The final grade (if the total number of points is at least 50) is given by applying the ECTS rules.
**COURSE NAME**
MULTIMEDIA TECHNOLOGIES IN AUTOMOTIVE

**CODE:** MISS1206

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>MASTER 1</th>
<th>SEMESTER</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COURSE STATUS</strong></td>
<td>(C-compulsory/OP-optional/F-facultative)</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>P (during the semester, C-oral examination, E-written examination, M-mixed)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2020</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TOTAL HOURS PER SEMESTER</th>
<th>56</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL HOURS T</td>
<td>94</td>
</tr>
<tr>
<td>INDIVIDUAL ACTIVITY</td>
<td>8</td>
</tr>
</tbody>
</table>

**TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME**
EXPERT OF THE COMPANY
CONTINENTAL AUTOMOTIVE ROMANIA

**DEPARTMENT**
Computer Science

**OBJECTIVES**
Assimilating the basic information needed for creating a multimedia application in the automotive environment. Fundamental knowledge of the domain, workflow, procedures and quality standard required in the field.

**GENERAL DESCRIPTION**

**DESCRIPTION OF SEMINARY / LABORATORY WORKS**

**TEACHING METHODS**
Presentations for the courses. Workshops with FCS and CONTINENTAL AUTOMOTIVE ROMANIA provided software and hardware during the laboratories. Questions and answers with expert guests.

**BIBLIOGRAPHY**
(SELECTION)
Boris Beizer: *Software Testing Techniques*
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software*

**EVALUATION**

<table>
<thead>
<tr>
<th>conditions</th>
<th>Laboratory involvement and results (L) – 60%. Written examination for the courses (WE1, WE2) – 40%.</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td>More than 50p in total (maximum of 100p). L &gt; 30p, WE1, WE2 &gt; 10p.</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>L – the maximum 60p will be divided between the 14 laboratories. Evaluations will be made for each laboratory, taking into account the overall involvement (bonuses) and the results relative to the targets. Specific criteria will be provided before each laboratory. There will be no homework assignments.</td>
</tr>
<tr>
<td>WEs – the written tests will verify the level of understanding and assimilation for the information presented during the courses.</td>
<td></td>
</tr>
<tr>
<td>final result - formula</td>
<td>Final grade = (L + WE1 + WE2) / 10</td>
</tr>
</tbody>
</table>
**COURSE NAME**: SPECIAL CHAPTERS IN HUMAN-COMPUTER INTERACTION  
**CODE**: MISS1207

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C S L Pr.</td>
<td>2</td>
<td>56</td>
<td>124</td>
<td>8</td>
<td>Romanian</td>
</tr>
</tbody>
</table>

**TEACHING METHODS**

Interactive presentations. Direct interaction. Online access to additional resources via the Website course.

**BIBLIOGRAPHY**

19. * * *, *HCI Design Patterns*: http://www.hcipatterns.org/
20. * * *, *UsiXML Consortium*: http://www.usixml.org/

**EVALUATION**

- **conditions**: 1 project (P), 1 test during semester (T)
- **criteria**: project P>5, test T>5
- **evaluation methods**: 1 project (P), 1 test during semester (T)
- **final result - formula**: 0.5 * P + 0.4 * T + 1

**OBJECTIVES**

To provide a comprehensible vision of the user interface design aspects and the user-computer interaction types. The students will be able to design ergonomic and attractive interfaces for specific types of users, applications, platforms, and devices, including Web and mobile ones.

**GENERAL DESCRIPTION**

Definitions and terminology.  
Aspects regarding the human-computer interaction.  
User interface design: principles, models, and types. Design methodologies. Case studies: game development.  
Human factor. Usability.  

**DESCRIPTION OF SEMINARY / LABORATORY WORKS**

Human-computer interaction patterns. Issues on interaction and user-interface design. Specific user-interface prototyping. Techniques of user interface evaluation. User testing.

**PREVIOUS COURSES REQUESTED**

Software Engineering, Computer Graphics, Web Technologies
# SOFTWARE SECURITY

**Code:** MISS2103

<table>
<thead>
<tr>
<th>Study Year</th>
<th>Semester</th>
<th>Course Status (C-compulsory/OP-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Hours Per Week</th>
<th>Total Hours Per Semester</th>
<th>Total Hours Individual Activity</th>
<th>Credits</th>
<th>Evaluation</th>
<th>Teaching Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>C S L Pr.</td>
<td></td>
<td></td>
<td></td>
<td>(P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td>Romanian</td>
</tr>
<tr>
<td>2 - 2 - 2</td>
<td>56</td>
<td>184</td>
<td>8</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Teacher</th>
<th>Teaching and Scientific Degree, First Name, Last Name</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. GHEORGHE GRIGORAS</td>
<td></td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

**Previous Courses Requested:** No prerequisite required.

**Objectives:**
The course is an introduction in various programming-based methods for the development of security policies. Students will acquire experience with programming with various Java security-related packages and with access control modules in Linux.

**General Description:**
The topics covered are:
1. Access control in Java
2. The Java Security and Cryptography extensions
3. SELinux, a Linux module for access control
4. Jif, a package for controlling information flow in Java programs
5. JAAS, the Java Authentication and Authorization System

**Description of Seminary/Laboratory Work:**
Seminars and laboratories are grouped around the chapter currently discussed in the course. They aim to illustrate the topics of the chapter mainly by practical applications.

**Teaching Methods:**
On-line and blackboard presentation.

**Bibliography (Selection):**

**Evaluation:**
- Conditions
- Criteria
- Evaluation methods: 7 homeworks and a final exam.
- Final result - formula: 50% from the homeworks and 50% from the final exam.
## COURSE DESCRIPTION

<table>
<thead>
<tr>
<th>COURSE NAME</th>
<th>QUALITY SYSTEMS SOFTWARE</th>
<th>CODE: MISS2207</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDY YEAR</td>
<td>II</td>
<td></td>
</tr>
<tr>
<td>SEMESTER</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>COURSE STATUS</td>
<td>(C-compulsory/OP-optional/F-facultative)</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
<td>56</td>
<td>184</td>
<td>M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>PROF. DR. DOREL LUCANU</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEACHING LANGUAGE</td>
<td>Romanian</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

| PREVIOUS COURSES REQUESTED | Engineering software systems (programming Engineering) |

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
<th>Understanding the main elements that define the quality of software. Familiarization with the testing methods and analysis software.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERAL DESCRIPTION</td>
<td>What is the quality of software? Quality Engineering Quality Testing software systems (concepts, activities, techniques, case studies) beyond quality assurance test (code inspection, formal verification, fault tolerance to) improve the quality of quantification (monitoring, measuring, quality models, analysis and classification of defects ) systems analysis software.</td>
</tr>
<tr>
<td>DESCRIPTION OF SEMINARY / LABORATORY WORKS</td>
<td>Working for individual projects that reflect the different activities to assure quality. Presentation of case studies. Reports.</td>
</tr>
<tr>
<td>TEACHING METHODS</td>
<td>Lectures on the system using interactive videoproieectorul, practical work</td>
</tr>
</tbody>
</table>

|--------------------------|----------------------------------------------------------------------------------------------------------|

<table>
<thead>
<tr>
<th>EVALUATION</th>
<th>conditions</th>
<th>Laboratory activity. Themes. Written exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>criteria</td>
<td></td>
<td>Each condition must be fulfilled to obtain at least a grade 5</td>
</tr>
<tr>
<td>evaluation methods</td>
<td>Laboratory work (40%); Themes 20%, written examination (40%), presentation of reports is a bonus.</td>
<td></td>
</tr>
<tr>
<td>final result - formula</td>
<td>Activity laboratory x 0.4 + theme x 0.2 + Exam x 0.4 + Reports</td>
<td></td>
</tr>
</tbody>
</table>

64
### Course Name
INTRODUCING NATURAL LANGUAGE PROCESSING

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>1</th>
<th>SEMESTER</th>
<th>2</th>
<th>COURSE STATUS (C-compulsory/OP-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOURS PER WEEK</td>
<td>2</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td></td>
</tr>
<tr>
<td>TOTAL HOURS PER SEMESTER</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>56</td>
</tr>
<tr>
<td>TOTAL HOURS INDIVIDUAL ACTIVITY</td>
<td>124</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREDITS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVALUATION</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td>M</td>
</tr>
<tr>
<td>(P-during the semester, C-oral examination, E-written examination, M-mixed)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEACHING LANGUAGE</td>
<td>Romanian</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Course Teacher
PROF. DR. DAN CRISTEA, PROF. DR. DAN TUFIS

### Previous Courses Requested
- 

### Objectives
The course of human language technologies to master students, deepening the knowledge taught in the course of Artificial Intelligence in the third year, in the direction of natural language processing and computational linguistics. Among the knowledge that circumscribes this vast area at the intersection between linguistics and computer science, to discuss ongoing problems is the understanding of semantic content of expression in natural language and speech analysis, the ultimate goal being the construction of models for automatic interpretation of texts.

### General Description
Classical theories of discourse (alert states theory, theory of rhetorical structure theory centers), problems of cohesion text (anaphoric and its resolution), stringers theory, incremental parsing text, summarizing text, ontologies language applications.

### Description of Seminary / Laboratory Works
Closely follows the concepts taught in class. Theme laboratories are posted at [http://profs.info.uaic.ro/~ipistol/tlu0708](http://profs.info.uaic.ro/~ipistol/tlu0708)

### Teaching Methods
Power Point presentations

### Bibliography (Selection)

### Evaluation

<table>
<thead>
<tr>
<th>Conditions</th>
<th>The minimum requirement for entry into the examination: laboratory 30 points (from 36) + 50 points project (of 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria</td>
<td>The minimum requirement for passage: 26 points 50 points laboratory + + Project 50 points sentence (of 100)</td>
</tr>
<tr>
<td>Evaluation Methods</td>
<td>Laboratories: 12 * (● present ● ● solved year, ● ● ● noted ) → max 36; Project: 0 - 100; written exam: 0 - 100</td>
</tr>
<tr>
<td>Final Result - Formula</td>
<td>Final Note: (100/36 * 1.1 * 1.2 * lab pro + ex) / 30, corrected by Gauss curve</td>
</tr>
</tbody>
</table>

---

65
COURSE NAME  |  NATURAL LANGUAGE PROCESSING BY STATISTICAL METHODS  |  CODE: ML1207
---|---|---
STUDY YEAR  |  I  |  SEMESTER  |  2  |  COURSE STATUS (C- compulsory/OP- optional/ F- facultative)  |  C
CLASS  |  MASTER OF COMPUTATIONAL LINGUISTICS, 2008 - 2010
HOURS PER WEEK  |  |  TOTAL HOURS PER SEMESTER  |  56  |  TOTAL HOURS INDIVIDUAL ACTIVITY  |  184  |  CREDITS  |  8  |  EVALUATION (P- during the semester, O- oral examination,  W- written examination, M- mixed)  |  M
TEACHING LANGUAGE  |  Romanian
COURSE TEACHER  |  Teaching and Scientific Degree, First Name, Last Name  |  Department
|  PROF. DR. DAN TUFIȘ  |  Computer Science
PREVIOUS COURSES REQUESTED  |  Probability and Statistics – CS 1210  |  Formal Languages, Automata and Compilers – CS 2103  |  Artificial Intelligence – CS 3103
OBJECTIVES  |  The introduction, development and deepening of some elements of linguistic and statistical corpora processing  |  Deepening the modalities of representation and manipulation of linguistic knowledge  |  Development of modeling capacities for building domain applications
DESCRIPTION OF SEMINAR / LABORATORY WORKS  |  Development, annotation and mark-up standards for corpora  |  Statistical processings and tests on corpora  |  Mono- and multi-lingual annotation of corpora for the lexical, morpho-syntactic, semantic, and discourse level
TEACHING METHODS  |  Interactive presentations combined with free discussions and debates
EVALUATION  |  conditions  |  Achieving homeworks and the semestrial project  |  Participation in the laboratory hours
|  criteria  |  Minimal achieving the homework and the semestrial project  |  Active participation in the laboratory classes
|  evaluation methods  |  Final written exam (60%)  |  Homeworks + project (40%)
|  final result - formula  |  \( NF = 0.6 \times E + 0.4 \times R + B \)  |  Where E = Exam grade, R = grade for homeworks and project, B – bonus for outstanding activity
COURSE NAME: APPLIED CRYPTOGRAPHY  
CODE: MSI1104

STUDY YEAR | 1 | SEMESTER | 1 | COURSE STATUS (C-compulsory/OP-optional/F-facultative) | C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION (P-during the semester, C-oral examination, E-written examination, M-mixed)</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

COURSE TEACHER  
TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME: PROF. DR. FERUCIO LAURENȚIU ȚIPEA, LECT. DR. SORIN IFTENE  
DEPARTMENT: Computer Science

PREVIOUS COURSES REQUESTED: Familiarity with basic cryptographic concepts as treated, for example, in the core course "Information Security", is helpful, but can in principle also be acquired in parallel to attending the course.

OBJECTIVES: The objectives of the course are to provide students with the practice of cryptographic algorithms and protocols (efficient implementations, vulnerabilities, etc.) and to provide a background against which the student will be able to assess existing applications of cryptography and develop new protocols and applications that employ cryptography.

GENERAL DESCRIPTION: Topics include:
- one-way functions, pseudo-randomness, hash functions
- symmetric encryption and authentication systems
- public-key encryption systems and PKI
- digital signature schemes
- cryptanalytic protocols
- crytopanalytic techniques
- applications and case studies

DESCRIPTION OF SEMINARY/LABORATORY WORK: During the course, students will be required to complete a project in applied cryptography. These projects must be completed in small groups (no more than 3 students) and each group will be required to present their progress to the other students twice during the course.

TEACHING METHODS: On-line and blackboard presentation.

BIBLIOGRAPHY (SELECTION):
- F.L.Tiplea: Introduction to Cryptography (in preparation) - chapters of the book will be available to students.

EVALUATION: One project and a final exam.

conditions
criteria
evaluation methods: One project and a final exam.
final result - formula: 50% from the project and 50% from the final exam.
### COURSE NAME
MODELS OF SECURITY

### CODE: MSI1205

### STUDY YEAR | SEMESTER | COURSE STATUS (C-compulsory/O-optional/F-facultative)
--- | --- | ---
1 | 2 | C

### HOURS PER WEEK

<table>
<thead>
<tr>
<th></th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### COURSE TEACHER

<table>
<thead>
<tr>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. CATALIN DIMA</td>
<td>LAACL (Univ. Paris 12, France)</td>
</tr>
</tbody>
</table>

### PREVIOUS COURSES REQUESTED

No prerequisite required.

### OBJECTIVES

The course is an advanced introduction in the modeling, analysis and validation of security properties and systems. Students will achieve basic and intermediate-level knowledge of various security analysis techniques, rating from access control models to information flow models and models for security protocols ensuring confidentiality, authenticity, anonymity and/or game-like adversial situations like fair exchange and auctions.

### GENERAL DESCRIPTION

The topics covered are:

- A revision of classical access control models: discretionary/mandatory, role-based, type enforcement, etc.; specification of access control properties.
- Models of information flow control: Bell-LaPadula, lattice models, models based on observability and interference, etc.
- Specification of information flow properties in extensions of temporal logics.
- Specifying information flow in programming languages, the Denning approach.
- Models for security protocols: tree automata, multi-agent models, coalitions and game semantics, models for trust management.
- Specification of various security properties (authenticity, anonymity, adversial and coalization situation) in extensions of temporal logics.
- Computational models of security protocols.

### DESCRIPTION OF SEMINARY / LABORATORY WORK

Seminars and laboratories are grouped around the chapter currently discussed in the course. They aim to illustrate the topics of the chapter mainly by practical applications.

### TEACHING METHODS

On-line and blackboard presentation.

### BIBLIOGRAPHY (SELECTION)

- Research papers on specific topics will be distributed during the lectures.

### EVALUATION

<table>
<thead>
<tr>
<th>CONDITIONS</th>
<th>CRITERIA</th>
<th>EVALUATION METHODS</th>
<th>FINAL RESULT - FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>7 homeworks and a final exam.</td>
<td>50% from the homeworks and 50% from the final exam.</td>
</tr>
</tbody>
</table>
PREVIOUS COURSES REQUESTED: Familiarity with the basic cryptographic concepts is helpful, but can in principle be also acquired in parallel to attending the course.

OBJECTIVES: Provide students with an understanding of the themes and challenges of network security, the techniques for access control and intrusion detection, and the current state of the art. The students will have developed a critical approach of the analysis of network security, and will be able to bring this approach to bear on future decisions regarding network security. Practical skills will include the implementation of a security protocol.

GENERAL DESCRIPTION: The course covers four main topics:
6. Authentication applications
7. IP security
8. Electronic mail security
9. Web security

DESCRIPTION OF SEMINARY / LABORATORY WORK: Students will be required to complete a project in network security. These projects must be completed in small groups (no more than 3 students) and each group will be required to present their progress to the other students twice during the course.

TEACHING METHODS: On-line and blackboard presentation.

BIBLIOGRAPHY (SELECTION):
- Research articles and RFCs on IPsec, SSL-TLS, DNSsec, etc.

EVALUATION:
- conditions
- criteria
- evaluation methods: One project and a final exam.
- final result - formula: 50% from the project and 50% from the final exam.
COURSE NAME: SECURITY PROTOCOLS

STUDY YEAR: 1  SEMESTER: 2  COURSE STATUS (C-compulsory/op-optional/f-facultative): C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C  S  L  Pr.</td>
<td>2 - 2 - 56</td>
<td>184</td>
<td>8</td>
<td>M Romanian</td>
</tr>
</tbody>
</table>

PREVIOUS COURSES REQUESTED: Familiarity with basic cryptographic concepts as treated, for example, in the core course "Information Security”, is helpful, but can in principle also be acquired in parallel to attending the course.

OBJECTIVES: The objectives of the course are to provide students with the practice of cryptographic protocols (efficient implementations, vulnerabilities, etc.) and to provide a background against which the student will be able to develop new protocols and applications.

GENERAL DESCRIPTION: Topics include:
- basics on security (cryptographic) protocols
- formalisms: MSR, strand spaces
- undecidability of secrets
- bounded protocols
- tagged protocols
- recursive protocols
- verification techniques: BAN, inductive method, strand spaces, based on I/O-automata

DESCRIPTION OF SEMINARY / LABORATORY WORK: All seminars will be oriented on the topic discussed during the courses. Students will be asked to prepare a research project.

TEACHING METHODS: On-line and blackboard presentation.

BIBLIOGRAPHY (SELECTION):
- F.L.Tiplea: Introduction to Cryptography (in preparation) - chapters of the book will be available to students.
- Research articles.

EVALUATION:
- conditions
- criteria
- evaluation methods: One project and a final exam.
- final result - formula: 50% from the project and 50% from the final exam.
### OBJECTIVES
The course is an introduction in various programming-based methods for the development of security policies. Students will acquire experience with programming with various Java security-related packages and with access control modules in Linux.

### GENERAL DESCRIPTION
The topics covered are:
- 10. Access control in Java
- 11. The Java Security and Cryptography extensions
- 12. SELinux, a Linux module for access control
- 13. Jif, a package for controlling information flow in Java programs
- 14. JAAS, the Java Authentication and Authorization System

### DESCRIPTION OF SEMINARY / LABORATORY WORK
Seminars and laboratories are grouped around the chapter currently discussed in the course. They aim to illustrate the topics of the chapter mainly by practical applications.

### TEACHING METHODS
On-line and blackboard presentation.

### BIBLIOGRAPHY (SELECTION)

### EVALUATION
- **conditions**
- **criteria**
- **evaluation methods**: 7 homeworks and a final exam.
- **final result - formula**: 50% from the homeworks and 50% from the final exam.
### Objectives
The course will present the most important mechanisms dedicated to protect data integrity and confidentiality, access control, authentication, user privacy, quality and continuity of service in wireless environments.

### General Description
The topics of the course are:
- Wireless technology overview
- Risks and threats of wireless
- Security under resource constraints (bandwidth, memory, computation, energy constraints)
- Intrusion and anomaly detection in wireless environments
- Key management in wireless environments
- Privacy and anonymity in wireless environments
- Public Key Infrastructure in wireless environments
- Authentication, authorisation, and access control in wireless environments
- Standards in wireless security (Equivalent Privacy Standard (WEP), Extensible Authentication Protocol (EAP), Wi-Fi Protected Access (WPA, WPA2), IEEE 802.11i, Bluetooth 2.1)
- Secure mobile commerce
- Secure wireless multimedia broadcast

### Description of Seminar/Laboratory Work
Seminars intend to stimulate students in their own research in wireless security - improve their ability of extracting, presenting, and discussing results from the most relevant papers and try to extend/improve them.

### Teaching Methods
On-line and blackboard presentation.

### Bibliography (Selection)
- NIST Federal Information Processing Standards.
- Conference and journal articles.

### Evaluation
| Conditions | Presentation of a report on a selected topic (P), midterm exam (ME), final exam (FE) |
| Criteria   | P, ME, FE ≥ 5 |
| Evaluation methods | Presentation of a report on a selected topic (P), midterm exam (ME), final exam (FE) |
| Final result-formula | 0.4 P + 0.3 ME + 0.3 FE |
COURSE NAME: SECURITY OF OPERATING SYSTEMS  
CODE: MSI2104

STUDY YEAR | II  
SEMESTER | 1  
COURSE STATUS (C-compulsory/O OPTIONAL/F-facultative) | C

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>28</td>
<td>62</td>
</tr>
</tbody>
</table>

COURSE TEACHER | LECT. DR. CRISTIAN VIDRASCU  
DEPARTMENT | Computer Science

PREVIOUS COURSES REQUESTED | BSc in Computer Science

OBJECTIVES | This course offers an advanced introduction in the field of security of operating systems. The students who will attend this course will obtain knowledge about operating systems, regarding the techniques and mechanisms used for their security.

GENERAL DESCRIPTION | The course will treat the following subjects:
- Access control in the system. Policies used for access control: discretionary, mandatory, and role-based policies. Specification of access control policies.
- The UNIX/Linux operating system. Security architecture. Classical access control mechanisms. Audit logs and intrusion detection methods.
- Advanced access control techniques - SELinux framework, AppArmor, Flask security architecture.
- Security evaluation standards: Orange Book, Common Criteria, etc.
- Modern protection mechanisms: virtualization and sandboxing techniques, proof-carrying code.

DESCRIPTION OF SEMINARY/LABORATORY WORK | This is a research oriented course aiming to guide students in doing research in the field of security of operating systems.

TEACHING METHODS | Exposure using video-projetor, combined with explanations on blackboard and practical demos.

• Research papers on specific topics.

EVALUATION | Practical labworks during the semester and final written test.
final result - formula | Lab * 0.4 + WrittenThesis1 * 0.3 + WrittenThesis2 * 0.3
# MALICIOUS SOFTWARE

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>II</th>
<th>SEMESTER</th>
<th>1</th>
<th>COURSE STATUS (C-compulsory/O-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>28</td>
<td>62</td>
</tr>
</tbody>
</table>

- Romanian

**COURSE NAME**
MALICIOUS SOFTWARE
**CODE:** MSI2104

**STUDY YEAR**
II
**SEMESTER**
1
**COURSE STATUS**
C

**PREVIOUS COURSES REQUESTED**
No requested prerequisite.

**OBJECTIVES**
This course aims to introduce students to the theory of malicious software (malware) such as computer viruses, worms, trojan horses, rootkits, spyware, dishonest addware, crimeware and other malicious and unwanted software. Defense techniques, detection methods and cleaning technologies are also discussed.

**GENERAL DESCRIPTION**
The course covers the following topics:
1. Introduction to malicious software (malware). Evolution of malware.
5. Trojan horses and their role in informatic attacks.

**DESCRIPTION OF SEMINARY / LABORATORY WORK**
Research oriented on malicious software design.

**TEACHING METHODS**
On-line and blackboard presentation.

**BIBLIOGRAPHY (SELECTION)**
- Research articles and software.

**EVALUATION**
<table>
<thead>
<tr>
<th>conditions</th>
<th>criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>evaluation methods</td>
<td>One project and final exam.</td>
</tr>
<tr>
<td>final result - formula</td>
<td>50% from the project + 50% from the final exam.</td>
</tr>
</tbody>
</table>

**DEPARTMENT**
Computer Science

**TEACHER**
DR. CONSTANTIN ENEA
**COURSE NAME**
BELIEF LOGICS IN INFORMATION SECURITY

**CODE:** MSI2205

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEMESTER</td>
<td>2</td>
</tr>
<tr>
<td>COURSE STATUS (C-compulsory/O-optional/F-facultative)</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>56</td>
<td>184</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE TEACHER</th>
<th>TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROF. DR. CRISTIAN-DUMITRU MASALAGIU</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

**PREVIOUS COURSES REQUESTED**
Familiarity with the basic cryptographic concepts is helpful, but can in principle be also acquired in parallel to attending the course.

**OBJECTIVES**
The objectives of the course are to provide students with basic analysis techniques for security protocols based on modal and beliefs logic, and to provide a background against which the student will be able to use software tools such as Isabelle and TLA+.

**GENERAL DESCRIPTION**
The topics of the course include:
- logic: syntax, semantics, normal forms, decidability
- logical theories, proof system, reasoning
- modal logics
- belief logics: BAN and GNY
- applications and case studies in Isabelle and TLA+

**DESCRIPTION OF SEMINARY / LABORATORY WORK**
Seminars/laboratories will illustrate the concepts discussed during the course. Students will be also required to complete a project and to present their progress to the other students twice during the course.

**TEACHING METHODS**
On-line and blackboard presentation.

**BIBLIOGRAPHY (SELECTION)**
- Research papers on BAN and GNY logics (to be expliciteely mentioned during the course)

**EVALUATION**
- conditions: Any student has to prove that he/she actually assisted to the labs.
- criteria: Any mentioned activity has to be quoted.
- evaluation methods: One project, activity during laboratories, and a final exam.
- final result - formula: 50% from the project and activity during laboratories and 50% from the final exam.
# SECURITY OF ELECTRONIC COMMERCE

**COURSE NAME:** SECURITY OF ELECTRONIC COMMERCE  
**CODE:** MSI2206

<table>
<thead>
<tr>
<th>STUDY YEAR</th>
<th>SEMESTER</th>
<th>COURSE STATUS (C-compulsory/OO-optional/F-facultative)</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>2</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HOURS PER WEEK</th>
<th>TOTAL HOURS PER SEMESTER</th>
<th>TOTAL HOURS INDIVIDUAL ACTIVITY</th>
<th>CREDITS</th>
<th>EVALUATION</th>
<th>TEACHING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>S</td>
<td>L</td>
<td>Pr.</td>
<td>56</td>
<td>184</td>
</tr>
</tbody>
</table>

**TEACHING AND SCIENTIFIC DEGREE, FIRST NAME, LAST NAME:** PROF. DR. VICTOR PATRICIU

**DEPARTMENT:** Technical Military Academy, Bucharest

**PREVIOUS COURSES REQUESTED:** Facultative: Applied Cryptography

## OBJECTIVES
This course introduces students to the challenge of electronic commerce and Business on Internet, vulnerabilities and defenses.

## GENERAL DESCRIPTION
The course covers the following topics:
1. Digital signatures and certificates and PKI
2. Smart-cards, biometrics and payment systems
3. E-commerce security
4. Regulations in e-commerce

## DESCRIPTION OF SEMINAR/LABORATORY WORK
PGP, cryptographic libraries (BSAFE, OpenSSL, MS CAPI/CAPICOM, Cryptolib, Java Cryptography), SET (Secure Electronic Transaction), iKP (Internet Keyed Payments), eCash, NetCash. Students will also be involved in writing a Java Card application for a dual digital signature under SET.

## TEACHING METHODS
On-line and blackboard presentation.

## BIBLIOGRAPHY (SELECTION)

## EVALUATION
<table>
<thead>
<tr>
<th>conditions</th>
<th>criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>evaluation methods</td>
<td>7 small projects and a final exam.</td>
</tr>
<tr>
<td>final result - formula</td>
<td>50% from the projects + 50% from the final exam.</td>
</tr>
</tbody>
</table>
III. OTHER INFORMATION ABOUT THE FACULTY

1. Rooms
   - 5 lecture rooms
   - 4 seminary rooms
   - 8 laboratories (20-25 computers/room)

2. Student associations
   The Association of Computer Science Students in Iași (ASII)
   - E-mail: asii@infoiasi.ro
   - Web: http://students.infoiasi.ro/~asii

3. Industry partnerships
   The Faculty of Computer Science has currently developed partnerships with important IT firms, such as:
   - Siemens Romania, who offers every year a certain amount of student scholarships for carrying out the practice stage within the firm
   - Microsoft Romania
   - Continental VDO (former Siemens VDO) Romania
   - DiTech Romania
   - Embarcadero Romania
   - Code40
   - Synygy

4. Student facilities
   The library
   The students of the Faculty of Computer Science have access to both the Central University Library "Mihai Eminescu" and its faculty branch (i.e., the Library of the Faculty of Computer Science).
   The collection of the Library of the Faculty of Computer Science comprises more than 2000 volumes (books and journals) in the fields of computer science and mathematics. Besides, the library grants access, on an online reservation basis, to the collection of around 100 installation and documentation CDs provided by Microsoft Romania.
   Internet access
   The students of the Faculty of Computer Science have free Internet access from the faculty's laboratories. These are open on weekdays (Monday to Friday), between 8-20, and are meant both for seminary/laboratory classes and for students' individual training.
   Internet access is also available from the campuses of the "Alexandru Ioan Cuza" University.
   Accommodation
   For students who do not live in Iași, the "Alexandru Ioan Cuza" University provides accommodation in its campuses: Titu Maiorescu, Codrescu, Târgușor Copou.
   Scholarships
   The scholarship system applied by the Faculty of Computer Science complies with the specific regulations of the "Alexandru Ioan Cuza" University. The main categories are study and performance scholarships (granted to students with excellent learning results) and social support scholarships (granted to students with lower income). There are also a series of scholarship categories that encourage and reward various kinds of performance.