(1) GENERAL

<table>
<thead>
<tr>
<th>SCHOOL</th>
<th>Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACADEMIC UNIT</td>
<td>Department of Computer Engineering and Informatics</td>
</tr>
<tr>
<td>LEVEL OF STUDIES</td>
<td>Undergraduate</td>
</tr>
<tr>
<td>COURSE CODE</td>
<td>CEID_NY240</td>
</tr>
<tr>
<td>SEMESTER</td>
<td>Spring</td>
</tr>
<tr>
<td>COURSE TITLE</td>
<td>Numerical Analysis and Implementation Environments</td>
</tr>
</tbody>
</table>

**INDEPENDENT TEACHING ACTIVITIES**

<table>
<thead>
<tr>
<th>WEEKLY TEACHING HOURS</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lectures, Tutorials, Laboratory</td>
<td>3 (L), 1 (T), 1 (L)</td>
</tr>
</tbody>
</table>

**TOTAL** 5

**COURSE TYPE**

Special background, skills development

**Q**

Recommended prerequisite knowledge: Good familiarity with the courses “Mathematics I”, “Mathematics II” (NY101), “Linear Algebra” (NY110), “Introduction to Algorithms” (NY205) or equivalents.

**LANGUAGE OF INSTRUCTION and EXAMINATIONS:**

Greek (the course may be offered in English for Erasmus students).

**IS THE COURSE OFFERED TO ERASMUS STUDENTS**

Yes

**COURSE WEBSITE (URL)**

https://eclass.upatras.gr/courses/CEID1066/

(2) LEARNING OUTCOMES

**Learning outcomes**

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area
- Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B
- Guidelines for writing Learning Outcomes

**Upon conclusion of the course the students ought to be able:**

1. To be familiar with the basic principles of floating-point arithmetic, its differences from classical arithmetic and to understand that mathematical equivalence does not imply equivalence in computational results and the applications thereof. They should also be familiar with the basic classes of numerical problems (direct, inverse, identification) and how the rank in terms of difficulty.
2. To be familiar with the basic types of numerical errors affecting problem solving and the basic principles governing the propagation of roundoff errors in floating-point.
3. To be familiar with the differences between direct and iterative methods for the numerical solution of linear systems of algebraic equations. To be familiar with the basic matrix factorizations and their applications in the numerical solution of linear systems including least squares problems.
4. To be familiar with the basic principles of iterative solution methods and their convergence properties.
5. To be familiar with variants of the power method and the basic idea behind the QR algorithm for the numerical solution of the algebraic eigenvalue problem.
6. To be familiar with the reasons polynomials are a basic tool for the approximation and interpolation of functions. To be familiar with the process of polynomial interpolation, related numerical difficulties and methods for addressing them.
7. To be familiar with the fundamental methods for solving nonlinear equations.
8. To be familiar with finite differences for approximating derivatives.
9. To be familiar with basic methods of numerical integration/quadrature.
10. To be familiar how to convert the solution of an ordinary differential equation to a system of equations using finite differences.
11. To be familiar with the MATLAB or the Octave problem solving environment and to be familiar with the choices offered by any of these for addressing the previously listed problems.
Upon conclusion of the course the students are expected to have the following skills:

1. Designing and implementing algorithms for the MATLAB or the Octave problem solving environments and selecting an appropriate method for solving problems numerically from their libraries. They should also be familiar with the numerical methods on which these libraries are based.

2. Formulating numerical computations in ways that reduce the worst case roundoff error.

3. Describing and using the LU, Cholesky and QR factorizations for solving linear systems and least squares problems and to be able to implement simple versions thereof. They should also be familiar with the numerical methods on which these libraries are based.

4. Describing and implementing simple iterative methods (e.g. Jacobi, Gauss-Seidel and Richardson) and investigating and stating results about their convergence based on the properties of the underlying problem.

5. Designing and implementing versions of the power method (including the inverse power and shifted inverse power methods) for computing selected eigenpairs.

6. Computing an interpolation polynomial from given data and to select the most appropriate representation depending on the problem specifications.

7. Constructing simple versions of nonlinear solvers based on bisection, Newton and secant.

8. Writing approximations for first and second order derivatives of functions and to compute bounds for the discretization error.

9. Performing numerical integration by selecting from simple quadrature rules (rectangle, trapezoidal, Simpson).

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Adapting to new situations
- Decision-making
- Working independently
- Team work
- Working in an international environment
- Working in an interdisciplinary environment
- Production of new research ideas
- Production of free, creative and inductive thinking
- Respect for difference and multiculturalism
- Respect for the natural environment
- Showing social, professional and ethical responsibility and sensitivity to gender issues
- Criticism and self-criticism
- Production of free, creative and inductive thinking
- Others...

SYLLABUS

1. Introduction to Numerical Analysis.
2. Introduction and basic elements of MATLAB and Octave.
3. Representation of real numbers in finite precision and the floating-point model of arithmetic.
6. Iterative methods for solving linear systems (Jacobi, Gauss-Seidel, Richardson).
14. The role of Numerical Analysis in Computer Science and Engineering.
(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY
Face-to-face, Distance learning, etc.

Ex cathedra.

USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Use of ICT in teaching, laboratory education, communication with students

The course makes heavy use of the facilities offered by the e-Class environment. All course notes and transparencies are placed online, as well as problem sets, and pointers to the relevant literature.

TEACHING METHODS
The manner and methods of teaching are described in detail.

Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc.

The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS

<table>
<thead>
<tr>
<th>Activity</th>
<th>Semester workload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lectures</td>
<td>3*13=39</td>
</tr>
<tr>
<td>Tutorials (exercises)</td>
<td>1*13=13</td>
</tr>
<tr>
<td>Laboratory exercises</td>
<td>1*13=13</td>
</tr>
<tr>
<td>Individual study, preparation</td>
<td>3*13=39</td>
</tr>
<tr>
<td>and problem solving.</td>
<td></td>
</tr>
<tr>
<td>Weekend study</td>
<td>2*13=26</td>
</tr>
<tr>
<td>Study during the 3 “empty</td>
<td>4*3=12</td>
</tr>
<tr>
<td>weeks” (2 weeks of vacation</td>
<td></td>
</tr>
<tr>
<td>and 1 week of exam preparation).</td>
<td></td>
</tr>
<tr>
<td>Total (25-30 hours per ECTS unit).</td>
<td>142</td>
</tr>
</tbody>
</table>

STUDENT PERFORMANCE EVALUATION
Description of the evaluation procedure

Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, etc.

Specifically-defined evaluation criteria are given, and if and where they are accessible to students.

The language of instruction and examination is Greek. Special provisions (lecture notes and examinations in English) can be made for foreign students.

The final grade is based 100% on performance on the final examination.

The final examination is written, of graded difficulty, and can consist of multiple choice questions, questions for short and long essay answers, «mini-lab»-type questions.

During the course students are also assigned a number of practical and theoretical exercises. Their purpose is to let students familiarize themselves with the use and possibilities available by MATLAB and Octave for solving numerical problems as well as in interpreting and evaluating the results obtained.

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:
  - Cleve Moler, «Numerical Methods with MATLAB» (available online).
  - U. Ascher και Chen Greif, A first course in numerical methods, SIAM, 2011 d.
  - On-line guides for MATLAB and Octave.

- Relevant scientific journals:
  - This is an introductory course hence there is no systematic use of articles from the scientific literature even though presentations make reference to the recent literature mostly in order to demonstrate to students the relevance of the course for the state of the art in Computer Science and Engineering and their applications.