SYLLABUS

1. Data about the program of study

1.1	Institution	The Technical University of Cluj-Napoca
1.2	Faculty	Faculty of Civil Engineering
1.3	Department	Structural Mechanics
1.4	Field of study	Civil Engineering
1.5	Cycle of study	Bachelor of Science
1.6	Program of study/Qualification	Civil Engineering (english)/Civil Engineer
1.7	Form of education	Full time
1.8	Subject code	4.00

2. Data about the subject

2.1	Subject name			Computer Programming and Design Applications (I)			
2.2	Subject area			Civil Engineering			
2.3	Course responsible/lecturer			Assoc.prof. F.-Zsongor GOBESZ			
2.4	Teachers in charge of seminars			Assoc.prof. F.-Zsongor GOBESZ			
2.5 Year of study	1	2.6 Semester	1	2.7 Assessment	C	2.8 Subject category	DF / DI

3. Estimated total time

3.1 Number of hours per week	4	3.2 of which, course:	2	3.3 applications:	2
3.4 Total hours in the curriculum	56	3.5 of which, course:	28	3.6 applications:	28
Individual study					hours
Manual, lecture material and notes, bibliography					12
Supplementary study in the library, online and in the field					3
Preparation for seminars/laboratory works, homework, reports, portfolios, essays					3
Tutoring					–
Exams and tests					1
Other activities					–

3.7	Total hours of individual study	19
3.8	Total hours per semester	75
3.9	Number of credit points	3

4. Pre-requisites (where appropriate)

4.1	Curriculum	none
4.2	Competence	none

5. Requirements (where appropriate)

5.1	For the course	Classroom with blackboard, videoprojector and screen
5.2	For the applications	Labroom with PCs, videoprojector and screen

6. Specific competences

Professional competences	After completing the syllabus, the students will be able to: - use the MS Windows operating system (individually and sharing resources in LAN); - create and handle electronic documents (word processing, spreadsheet etc.); - describe an algoritm through logical scheme (by procedural reasoning); - develop or modify Fortran console applications using development toolkits (CVF, Force2 etc.); - use external mathematical libraries for engineering calculus; - transfer data through network, or by using network storage or external drives; - use the MS Office package, the Compaq Visual Fortran SDK and Force2 PE.
Cross competences	Knowledge and experience of employing efficient and responsible work strategies, punctuality, seriousness and liability based on the principles, norms and values of professional ethics. Applying efficient technics in team work. Development of self-expression, vocabulary and technical culture. Professional and personal development through continuous training and active adaptation to new technical specifications.

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

7.1	General objective	To develop skills in applied informatics and to improve deterministic way of thinking by procedural approaches.
7.2	Specific objectives	Assimilation of theoretical and practical knowledge about the use of computers and the development of Fortran applications.

8. Contents

8.1. Lecture (syllabus)		Teaching methods	Notes
1.	Introduction. Overview, objectives, how to conduct the discipline. Brief history of computer equipment and information technology, fundamental concepts, hardware, evolutionary aspects.	Oral and written presentation with examples and comments (stimulating interactivity)	Individual study topics will be announced a week before
2.	Operating systems. Concepts, developments and trends, main components and functions. Physical and logical aspects of storing and managing data. File specifiers and usual formats.		
3.	Algorithms, methods and descriptive tools. Concepts, method classifications and paradigms. Task analysis and software development stages. Analysis and method description tools. Structuring concepts. Types of errors in data collection and automatic data processing.		
4.	Software development in Fortran language. Brief history of the language, the steps of creating a computer program, types of applications. Vertical and horizontal structure of a Fortran source file, existing standards, program units. Accessible resources and environments for software development, issues concerning licenses and copyrights. Used set of characters, marks, comments, symbolic names.		
5.	Data types in Fortran. Types of entities, the default rule. Statements, characteristics and attributes,		

	statements and data structures. Arrays. User-defined data types, structured specifications.		
6.	Expressions in Fortran. Composition of numerical and logical expressions, types of operators and operands, evaluation mode and priorities, compatibility of types and conversions. Operator overloading.		
7.	Jump and control instructions. Labels, Instruction types, structured and accepted unstructured versions, notions of code optimization.		
8.	Input-output instructions. Input / output operations and instructions, options. Logical units, file organization and access, usual parameters.		
9.	Format specifiers, editing and control descriptors, examples.		
10.	Fortran program units. Main program. External (and internal) procedures, specification and use, entry and return/exit points, transfer of values. Block Data units.		
11.	Modules. Explicit interfaces, module procedures. Advanced topics: Visibility of entities. Dynamic memory allocation. Pointers and arrays of pointers.		
12.	Using mathematical libraries, useful examples from IMSL and CXML for civil engineering. Resources available through the internet. Language evolution: Concepts of object-oriented programming in Fortran. Differences between the F90 / F95 and Fortran 2003/2008, HPF. Possibilities of interconnection with other programming languages, examples.		
13.	Computer and communication networks. Brief history, basic topologies, shared resources, protocols, types of computer networks. Security issues.		
14.	BIM & PDT. Concepts and standardization in civil engineering, interoperability, concurrence. XML applications in architecture, engimeering and construction. *Written test of the theory*.		

Bibliography

1. Lecture notes.
2. *Basic Computing Using Windows*, Wikibooks.org, 2006.
3. *Compaq Fortran, Language Reference Manual*, Compaq Computer Corporation, Houston, Texas, 1999.
4. Dijkstra, E., D.: *Notes on Structured Programming*, Second Edition T. U. Eindhoven, 1970.
5. Jorgensen, E.: Introduction to Programming using Fortran 95/2003/2008, University of Nevada, Las Vegas, 2013
6. Marshall, A., C. - Morgan, J., S. - Schonfelder, J., L.: Fortran 90 Course Notes, The University of Liverpool, UK, 1997.
7. Sandu, A.: Introduction to Fortran 95 and Numerical Computing, Virginia Tech, 2001.
8. http://buildingsmart.org (Open BIM, IFC)
9. http://groups.engin.umd.umich.edu/CIS/course.des/cis400/fortran/fortran.html (The Fortran Programming Language)
10. http://www.fortran.com

11. http://www.mathtools.net
12. http://users.utcluj.ro/~go/ (handouts and further resources)

8.2. Applications/Seminars		Teaching methods	Notes
1.	Presentation of the laboratory and of the equipment, health and safety isues, rules of conduct, organizational aspects. How to use the equipment and peripherals, available resources. Expected portfolio and assessment.	Short presentation, examples and solutions with discussion, followed by individual subjects for each student	Each student has to work on a PC, the solved topics will be checked and assessed weekly by the teacher
2.	Office applications, creating an electronic document, settings, processing and formatting issues. Embedded objects, links and interconnectivity. Expressions, charts, drawings, images and references included in a document. Conversion between common electronic formats.		
3.	Use of spreadsheets, layouts, calculus, charts, embedded objects and links, references.		
4.	Flowcharts. Use of structural primitives, exercises.		
5.	Translation of numerical and logical expressions in Fortran, exercises. Basics of translating structured flowcharts in Fortran.		
6.	The GUI of CVF and Force2 SDK. Creating a Fortran console application from a previous flowchart. Compiler options, handling compiling and link-editing errors and warnings, trace and debug.		
7.	Exercises with vector arrays and strings (extreme values and sorting methods).		
8.	Exercises with matrix arrays (transpose, multiplication, use of conformance).		
9.	Exercises with vector and matrix arrays using dynamic memory allocation and implied loops, I/O using files.		
10.	Exercises with procedures (subroutines and functions).		
11.	Exercises with user-defined data types and arrays, using subroutines and functions.		
12.	Matrix calculus using IMSL (external mathematical library).		
13.	Exercises with pointers (input / output string handling).		
14.	*Practical test*. Portfolio rating and discussion of the work done during the semester.		

Bibliography

1. Class notes and handouts.
2. *Compaq Visual Fortran, Language Reference Manual*, Compaq Computer Corporation, Houston, Texas, 1999.
3. Lepsch, G.: Force Fortan – The Force Project (http://force.lepsch.com/)
4. Rogue Wave: IMSL Numerical Libraries, *Fortran Library documentation* (http://www.roguewave.com/help-support/documentation/imsl-numerical-libraries#fortran).
5. http://users.utcluj.ro/~go/ (samples and further resources)

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

Acquired skills will be needed by engineers working in building design and/or research (also in education).

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
Course	Theory (9 questions from theory, against time)	Written test	40%
Applications	Activity during the semester (portfolio with solved problems) + Solving a problem in a given time	Assessment of each labwork Practical exam on computer	20% 40%
10.4 Minimum standard of performance			
Solving and handing over of labworks by deadlines and getting at least 4.5 points individually at each of the three assessment criteria.			

Date of filling in
18.07.2017

Teachers in charge of seminars
Assoc.Prof. F.-Zsongor GOBESZ

Date of approval in the department

Head of department
Prof. Cosmin G. CHIOREAN