SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babeş Bolyai University**
1.2 Faculty	**Faculty of Mathematics and Computer Science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Master**
1.6 Study programme / Qualification	**Applied Computational Intelligence**

2. Information regarding the discipline

2.1 Name of the discipline	**Framework Design**						
2.2 Course coordinator	**Lect. dr. Ioan Lazar**						
2.3 Seminar coordinator	**Lect. dr. Ioan Lazar**						
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Optional**

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	4	Of which: 3.2 course	2	3.3 seminar/laboratory	1+1
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					8
Additional documentation (in libraries, on electronic platforms, field documentation)					7
Preparation for seminars/labs, homework, papers, portfolios and essays					8
Tutorship					2
Evaluations					8
Other activities:					

3.7 Total individual study hours	119
3.8 Total hours per semester	175
3.9 Number of ECTS credits	7

4. Prerequisites (if necessary)

4.1. curriculum	• Programming Fundamentals
4.2. competencies	• Good programming skills in at least one of the languages Java, C#

5. Conditions (if necessary)

5.1. for the course	• Course hall with projector
5.2. for the seminar /lab activities	• Laboratory with computers

6. Specific competencies acquired

Professional competencies	• C 4.3 Identify models and methods adequate to real life problem solving • C 2.1 Identify adequate software systems development methodologies • C 1.1 Proper description of programming paradigms and language specific mechanisms, and identification of semantical an syntactical differences
Transversal competencies	• CT1 Apply organized and efficient work rules and responsible attitude towards didactical and research field, in order to creatively use work potential; respect professional ethical principles • CT3 Use efficient methods and techniques for: learning, information search, research and development of capacities to adapt to the requirements of a dynamic society and to communicate in an international language

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	Enhance the students understanding of service oriented concepts through a practical and pragmatic approach Provide the students with an environment in which they can explore the usage and usefulness of service oriented concepts in various business scenarios Induce a realistic and industry driven view of software design concepts such as design patterns and their inherent benefits
7.2 Specific objective of the discipline	Give students the ability to explore various object oriented programming languages Improve the students abilities to tackle business requirements Enhance the students understanding of business needs and business value Provide students with insights into the way of working towards achieving high quality software through skilled trainers from the IT industry

8. Content

8.1 Course	Teaching methods	Remarks
1. Web frameworks for Node.js PBD/Web Platforms Web programming languages - JavaScript - callback, generator, async functions	Exposure: description, explanation, examples, discussion of case studies	

SE/Software Design Web frameworks for node based on - callback functions - generator functions - async functions - reactive extensions (rxjs)		
2. Functional reactive programming (FRP) - pure functions, higher order functions - recursion - map, reduce, filter - functional composition	Exposure: description, explanation, examples, discussion of case studies	
3. Web frameworks based on FRP 3.1 HCI/Programming Interactive Systems Functional reactive programming - Cycle.js, https://cycle.js.org/	Exposure: description, explanation, examples, discussion of case studies	
4. Web frameworks based on FRP 4.1 HCI/Programming Interactive Systems Functional reactive programming - Recycle.js, https://recycle.js.org/	Exposure: description, explanation, examples, discussion of case studies	
5. Component based web frameworks Components - properties, lifecycle, state, and events - composition vs inheritance - Inferno.js, https://github.com/infernojs/inferno Application state - flux architecture	Exposure: description, explanation, examples, discussion of case studies	
6. Component based web frameworks Elements - properties and behaviors - composition - Polymer, https://www.polymer-project.org Application state - elements without UI	Exposure: description, explanation, examples, discussion of case studies	
7. Component based web frameworks Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/ Application state - services	Exposure: description, explanation, examples, discussion of case studies	

8. Creating a model-based framework for user interfaces IFML metamodel - domain model - services, actions - components, containers	Exposure: description, explanation, examples, discussion of case studies	
9. Creating an IFML diagram editor - components, containers - navigation flow	Exposure: description, explanation, examples, discussion of case studies	
10. Creating a domain model diagram editor - classes, properties, associations	Exposure: description, explanation, examples, discussion of case studies	
11. Running and deploying components - run component within the framework - generate code and run components as standalone apps	Exposure: description, explanation, examples, discussion of case studies	
12. Component repository - publish components - reuse components	Exposure: description, explanation, examples, discussion of case studies	
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Creating a secured server for component repositories	Dialogue, debate, case studies, examples, proofs	
2. Creating a web app based on FRP frameworks	Dialogue, debate, case studies, examples, proofs	
3. Creating a web app based on web components	Dialogue, debate, case studies, examples, proofs	
4. Creating a model-based framework for user interfaces	Dialogue, debate, case studies, examples, proofs	
5. Add diagram editors	Dialogue, debate, case studies, examples, proofs	
6. Add component repository features	Dialogue, debate, case studies, examples, proofs	

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curriculla Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;

• The content of the course is considered the software companies as important for average programming skills.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.5 Seminar/lab activities	Implement a system with REST services, server side notifications, and data synchronization	Project grading	100%
10.6 Minimum performance standards			
➢ A minimum passing grade is defined by attaining at least 50% (5/10) points for the final project and each of the three lab assignments respectively. ➢ No more than 3 absences are allowed for the seminar/lab activities			

Date	Signature of course coordinator	Signature of seminar coordinator
20.04.18	**Lect. dr. Ioan Lazar**	**Lect. dr. Ioan Lazar**

Date of approval	Signature of the head of department
	Prof. dr. Anca Andreica