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	1	2.5	2	2.6. Type of	Ε	2.7 Type of	Optional
study		Semester		evaluation		discipline	

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

3.1 Hours per week	4	Of which: 3.2	2	3.3	1+1
		course		seminar/laboratory	
3.4 Total hours in the curriculum	56	Of which: 3.5	28	3.6	28
		course		seminar/laboratory	
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					8
Additional documentation (in libraries, on electronic platforms, field documentation)					
Preparation for seminars/labs, homework, papers, portfolios and essays					8
Tutorship 2					
Evaluations					8
Other activities:					
3.7 Total individual study hours		119			1

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	Laboratory with computers
activities	

6. Specific competencies acquired

Profe ssion al comp etenc ies	 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
Tran svers al comp etenc ies	 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)

	Enhance the students understanding of service oriented concepts through a practical and pragmatic approach
7.1 General objective of the discipline	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	Exposure:	
	description,	
PBD/Web Platforms	explanation,	
Web programming languages - JavaScript	examples, discussion	
	of case studies	
- callback, generator, async functions		

SE/Software Design Web frameworks for node based on - - callback functions - - - generator functions - - - sayne functions - - - reactive extensions (xrjs) Exposure: - 2. Functional reactive programming (FRP) Exposure: - - pure functions, higher order functions - explanation, - recursion - explanation, - - functional composition - - - 3. Web frameworks based on FRP Exposure: - - - functional reactive programming of case studies - - - Cvete js, https://cvete.js.org/ - - - - 4. Web frameworks based on FRP Exposure: - - - - 4. Ht/Programming Interactive Systems explanation, examples, discussion of case studies -		
 callback functions generator functions async functions reactive extensions (rsjs) Functional reactive programming (FRP) explanation, explanation, examples, discussion asynchaction, explanation, examples, discussion examples, discussion asynchaction, examples, discussion explanation, examples, discussion of case studies e	SE/Software Design	
 callback functions generator functions async functions reactive extensions (rsjs) Functional reactive programming (FRP) explanation, explanation, examples, discussion asynchaction, explanation, examples, discussion examples, discussion asynchaction, examples, discussion explanation, examples, discussion of case studies e	Web frameworks for node based on	
generator functions async functions async functions async functions reactive extensions (xjs) Exposure: description, explanation, examples, discussion of case studies functional reactive programming of case studies cycle.js, https://cycle.js.org/ two frameworks based on FRP thtps://cycle.js.org/ two frameworks based on FRP thtps://cycle.js.org/ two frameworks based on FRP two frameworks composition vs inheritance - functional reactive programming for case studies two frameworks composition vs inheritance - Inferno.js, https://recycle.js.org/ two frameworks two frameworks composition vs inheritance - Inferno.js, https://github.com/inferno Application state - Polymer, https://www.polymer-project.org Application state - properties and behaviors - composition com		
- async functions - reactive extensions (rxjs) 2. Functional reactive programming (FRP) - pure functions, higher order functions - recursion - may, reduce, filter - functional composition 3. Web frameworks based on FRP Based of frameworks based on FRP Cycle, is, https://cycle.js.org/ 4. Web frameworks based on FRP Exposure: description, examples, discussion of case studies - Cycle, is, https://cycle.js.org/ 4. Web frameworks based on FRP Exposure: description, examples, discussion of case studies - Cycle, is, https://cycle.js.org/ 4. Web frameworks based on FRP Exposure: description, examples, discussion of case studies - Cycle, is, https://cycle.js.org/ 5. Component based web frameworks Exposure: description, examples, discussion of case studies - Recycle, is, https://cycle.js.org/ 5. Component based web frameworks Exposure: description, explanation, examples, discussion of case studies - Inferno js, htups://github.com/infernojs/inferno Application state - Polymer, https://www.polymer-project.org Application state - Polymer, https://www.polymer-project.org Application state - Polymer, https://www.polymer-project.org Application state - Polymer, https://angular.io/ Application state - properties and behaviors - composition Application state - properties and behaviors - composition - Angular 2, https://angular.io/ Application state	- callback functions	
- reactive extensions (rxjs)Exposure: description, explanation, examples, discussion of case studies- pure functions, higher order functions - recursion - map, reduce, filter - functional compositionexamples, discussion of case studies3. Web frameworks based on FRP 3.1 HCJ/Programming Interactive Systems - Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies- Incerno.js, https://pithub.com/infernojs/infernoExposure: description, explanation, e	- generator functions	
2. Functional reactive programming (FRP) Exposure: - pure functions, higher order functions explanation, - recursion explanation, - map, reduce, filter of case studies 3. Web frameworks based on FRP Exposure: 3. Web frameworks based on FRP Exposure: 4.1 HCVProgramming Interactive Systems explanation, Functional reactive programming of case studies - Cycle.js, https://cycle.js.org/ Exposure: 4. Web frameworks based on FRP Exposure: 4. Web frameworks based on FRP Exposure: 4. Web frameworks based on FRP Exposure: 4. HCVProgramming Interactive Systems explanation, Functional reactive programming of case studies - Corposition vs inheritance explanation, - norponents explanation, - accycle.js, https://recycle.js.org/ explanation, Components based web frameworks Exposure: description, explanation, - composition vs inheritance explanation, - Inferno.js, https://github.com/infernojs/inferno of case studies Application state explanation,	- async functions	
- pure functions, higher order functions - recursiondescription, explanation, examples, discussion of case studies- map, reduce, filter functional compositionExposure: description, examples, discussion of case studies3.1 HCI/Programming Interactive SystemsExposure: description, examples, discussion of case studiesFunctional reactive programmingof case studies- Cycle is, https://cycle.is.org/Exposure: description, examples, discussion of case studies4.1 HCI/Programming Interactive SystemsExposure: description, explanation, examples, discussion of case studiesFunctional reactive programming - Recycle is, https://recycle.is.org/of case studiesFunctional reactive programming - Recycle is, https://recycle.is.org/case studiesComponents - properties, lifecycle, state, and events - oroposition vs inheritanceExposure: description, examples, discussion of case studies- Intw architecture6. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - oroposition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studies- Component based web frameworksExposure: description, examples, discussion of case studies- Polymer, https://angular.io/Exposure: description, examples, discussion of case studies- properties and behaviors - composition - properties and behaviors - composition - Polymer, https://angular.io/Exposure: descri	- reactive extensions (rxjs)	
 pure functions, higher order functions recursion map, reduce, filter functional composition Web frameworks based on FRP S. Web frameworks based on FRP a. HCUProgramming Interactive Systems explanation, examples, discussion of case studies c. Cycle.js, https://cycle.js.org/ Web frameworks based on FRP explanation, examples, discussion of case studies c. Cycle.js, https://cycle.js.org/ Web frameworks based on FRP explanation, examples, discussion of case studies c. Cycle.js, https://cycle.js.org/ Web frameworks based on FRP Exposure: description, examples, discussion of case studies execycle.js, https://cycle.js.org/ S. Component based web frameworks Exposure: description, examples, discussion of case studies execycle.js, https://cycle.js.org/ S. Components properties, lifecycle, state, and events composition vs inheritance Inferno.js, https://gitub.com/infernojs/inferno Application state elements ecomposition Properties and behaviors ecomposition Properties and behaviors ecomposition Properties and behaviors ecomposition Properties and behaviors elements and modules properties and behaviors component based web frameworks Exposure: description, examples, discussion of case studies 	2. Functional reactive programming (FRP)	
- recursionexamples, discussion of case studies- map, reduce, filter functional composition-3. Web frameworks based on FRPExposure: description, examples, discussion of case studies3.1 HCJ/Programming Interactive Systemsexfamation, examples, discussion of case studiesFunctional reactive programmingof case studies- Cycle.js, https://cycle.js.org/-4. Web frameworks based on FRPExposure: description, explanation, examples, discussion of case studies1. HCJ/Programming Interactive SystemsExposure: description, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/-5. Components - properties, lifecycle, state, and events - composition vs inheritance - Intw architectureExposure: description, examples, discussion of case studies6. Component based web frameworksExposure: description, examples, discussion of case studies9. properties and behaviors - oromposition - properties and behaviors - composition - polymer, https://www.polymer-project.orgExposure: 	- pure functions higher order functions	
- map, reduce, filterof case studies- functional composition		-
- functional composition3. Web frameworks based on FRP3. HCl/Programming Interactive SystemsFunctional reactive programming- Cycle.js, https://cycle.js.org/4. Web frameworks based on FRP4. Web frameworks based on FRP4. HCl/Programming Interactive SystemsFunctional reactive programming- Cycle.js, https://cycle.js.org/4. HCl/Programming Interactive SystemsFunctional reactive programming- Recycle.js, https://recycle.js.org/5. Component based web frameworks- Recycle.js, https://recycle.js.org/5. Component based web frameworks- properties, lifecycle, state, and events- composition vs inheritance- Inferno.js, https://github.com/infernojs/infernoApplication state- flux architecture6. Component based web frameworksElements- properties and behaviors- composition- Polymer, https://www.polymer-project.orgApplication state- elements without UI- Component based web frameworksExposure:description,explanation,explanation,exploration state- component based web frameworksExposure:description,exploration state- component sand modules- properties and behaviors- composition- Angular 2, https://angular.jo/Application state		
3. Web frameworks based on FRP Exposure: 3.1 HCl/Programming Interactive Systems explanation, Functional reactive programming of case studies - Cycle.js, https://cycle.js.org/ explanation, 4. Web frameworks based on FRP Exposure: description, explanation, Functional reactive programming of case studies Functional reactive programming explanation, explanation, explanation, explore description, explore description, explore explanation, explanation, explanation, explore explore description, explanation, expo	-	
3.1 HCl/Programming Interactive Systemsdescription, explanation, examples, discussion of case studiesFunctional reactive programmingof case studies- Cycle.js, https://cycle.js.org/Exposure: description, examples, discussion of case studies4.1 HCl/Programming Interactive Systemsexplanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/Exposure: description, examples, discussion of case studiesS. Components - properties, lifecycle, state, and events - oromposition vs inheritanceExposure: description, explanation, examples, discussion of case studies<		Exposure:
3.1 HCl/Programming Interactive Systems explanation, Functional reactive programming of case studies - Cycle.js, https://cycle.js.org/ Exposure: description, explanation, 4.1 HCl/Programming Interactive Systems explanation, Functional reactive programming of case studies Functional reactive programming of case studies Functional reactive programming of case studies - Recycle.js, https://recycle.js.org/ Exposure: - Recycle.js, https://recycle, stare, and events explanation, - omposition vs inheritance explanation, - Infernojs, https://github.com/infernojs/inferno of case studies Application state - - flux architecture Exposure: 6. Component based web frameworks Exposure: elements explanation, - properties and behaviors examples, discussion - composition of case studies - Properties and behaviors examples, discussion - composition of case studies - Properties and behaviors examples, discussion - composition state explanation, <td< td=""><td>5. Web frame works based on FRI</td><td>-</td></td<>	5. Web frame works based on FRI	-
Functional reactive programmingexamples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies4. Web frameworks based on FRPExposure: description, explanation, examples, discussion of case studies4. I HCI/Programming Interactive SystemsExposure: description, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/of case studies5. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studies- Droporties, lifecycle, state, and events - composition vs inheritance - Inferno, js, https://github.com/infernojs/infernoexamples, discussion of case studies6. Component based web frameworksExposure: description, examples, discussion of case studies9. Component based web frameworksExposure: description, examples, discussion of case studies9. Properties and behaviors - composition - rompositionExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studies9. Differenciesetermine9. Component based web frameworksExposure: description, examples, discussion of case studies9. Component based web frameworksExposure: description, examples, discussion of c	3.1 HCI/Programming Interactive Systems	
Functional reactive programmingof case studies- Cycle.js, https://cycle.js.org/-4. Web frameworks based on FRPExposure: description, explanation, examples, discussion of case studies4.1 HCl/Programming Interactive Systems-Participation-Functional reactive programming - Recycle.js, https://recycle.js.org/-5. Component reactive programming - Recycle.js, https://recycle.js.org/-5. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studies- Droperties, lifecycle, state, and events - composition vs inheritance - Inferno.js, https://github.com/infernojs/inferno-Application state - flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies- Rux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies- Polymer, https://www.polymer-project.org-Application state - elements without UI - component based web frameworksExposure: description, examples, discussion of case studies- Rus and behaviors - composition - properties and behaviors - composition - properties and behaviors - composition - properties and behaviors - composition - properties and behaviors - composition - angular 2, https://angular.io/Exposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation	5.1 HCh rogramming interactive Systems	
- Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies4. Web frameworks based on FRPExposure: description, explanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/Exposure: description, explanation, examples, discussion of case studies5. Component based web frameworksExposure: description, explanation, <b< td=""><td>Eurotional reactive preservation</td><td></td></b<>	Eurotional reactive preservation	
4. Web frameworks based on FRP Exposure: description, explanation, examples, discussion of case studies Functional reactive programming of case studies - Recycle.js, https://recycle.js.org/ - 5. Component based web frameworks Exposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies - Inferno.js, https://github.com/infernojs/inferno - Application state - flux architecture - 6. Component based web frameworks Exposure: description, examples, discussion of case studies Elements - properties and behaviors - composition - composition examples, discussion of case studies Polymer, https://www.polymer-project.org of case studies Application state - elements without UI Exposure: description, examples, discussion of case studies 7. Component based web frameworks Exposure: description, examples, discussion of case studies Polymer, https://www.polymer-project.org Exposure: description, examples, discussion of case studies Application state - elements without UI Exposure: description, examples, discussion of case studies - properties and behaviors - composition - Angular 2, https://angular.io/ examples, discussion of case studies	runchonal reactive programming	of case studies
4. Web frameworks based on FRP Exposure: description, explanation, examples, discussion of case studies Functional reactive programming of case studies - Recycle.js, https://recycle.js.org/ - 5. Component based web frameworks Exposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies - Inferno.js, https://github.com/infernojs/inferno - Application state - flux architecture - 6. Component based web frameworks Exposure: description, examples, discussion of case studies Elements - properties and behaviors - composition - composition examples, discussion of case studies Polymer, https://www.polymer-project.org of case studies Application state - elements without UI Exposure: description, examples, discussion of case studies 7. Component based web frameworks Exposure: description, examples, discussion of case studies Polymer, https://www.polymer-project.org Exposure: description, examples, discussion of case studies Application state - elements without UI Exposure: description, examples, discussion of case studies - properties and behaviors - composition - Angular 2, https://angular.io/ examples, discussion of case studies	Cuala is https://augla.is.org/	
4.1 HCl/Programming Interactive Systemsdescription, explanation, examples, discussionFunctional reactive programming - Recycle.js, https://recycle.js.org/of case studies5. Component based web frameworksExposure: description, examples, discussion of case studies- properties, lifecycle, state, and events - composition vs inheritance - Inferno.js, https://github.com/infernojs/infernoexamples, discussion of case studiesApplication state - flux architectureexplanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesElements - properties and behaviors - composition - composition - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studiesApplication state - elements without UIExposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, examples, discussion of case studiesPolymer, https://www.polymer-project.orgApplication state - elements without UI7. Component based web frameworksExposure: description, explanati		Exposure
4.1 HCl/Programming Interactive Systems explanation, Functional reactive programming of case studies Functional reactive programming of case studies - Recycle.js, https://recycle.js.org/ Exposure: description, explanation, - properties, lifecycle, state, and events explanation, - properties, lifecycle, state, and events explanation, - composition vs inheritance of case studies - Inferno.js, https://github.com/infernojs/inferno of case studies Application state explanation, - flux architecture Exposure: description, explanation, - properties and behaviors explanation, - properties and behaviors explanation, - properties and behaviors explanation, - Polymer, https://www.polymer-project.org of case studies Application state elements - elements without UI explanation, - properties and behaviors explanation, -	4. web frameworks based on FKF	
Functional reactive programming - Recycle.js, https://recycle.js.org/examples, discussion of case studies5. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studiesComponents - properties, lifecycle, state, and events - composition vs inheritance - Inferno.js, https://github.com/infernojs/infernoexamples, discussion of case studiesApplication state - flux architectureExposure: description, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies9 roperties and behaviors - properties and behaviors - Component based web frameworksExposure: description, examples, discussion of case studies- Recycle is and modules - properties and behaviors - composition - properties and behaviorsExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studies9. Component based web frameworksExposure: 	4.1 HCI/Dro gramming Intersective Systems	
Functional reactive programmingof case studies- Recycle.js, https://recycle.js.org/-5. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, examples, discussion of case studies- Composition vs inheritance - Inferno.js, https://github.com/infernojs/inferno-Application state - flux architecture-6. Component based web frameworksExposure: description, explanation, examples, discussion of case studies- Component based web frameworksExposure: description, examples, discussion of case studies- Polymer, https://www.polymer-project.orgExposure: description, explanation, <b< td=""><td>4.1 HCI/Programming interactive Systems</td><td>-</td></b<>	4.1 HCI/Programming interactive Systems	-
- Recycle.js. https://recycle.js.org/ 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 6. Component based web frameworks Elements - properties and behaviors - composition - Polymer, https://www.polymer-project.org Application state - elements without UI 7. Component based web frameworks Exposure: description, explanation, e elements without UI 7. Component based web frameworks Exposure: description, explanation, e elements without UI 7. Component based web frameworks Exposure: description, explanation elements without UI 7. Component based web frameworks Exposure: description, explanation elements and modules enproperties and behaviors		
5. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studiesApplication state - flux architectureExposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, examples, discussion of case studiesElements - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studiesApplication state - elements without UIExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studiesPolymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studies7. Components and modules - properties and behaviors - composition - oropositionExposure: description, examples, discussion of case studies7. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies		of case studies
Components - properties, lifecycle, state, and events - composition vs inheritance - Inferno, js, https://github.com/infernojs/infernodescription, explanation, examples, discussion of case studiesApplication state - flux architectureExposure: description, explanation, examples, discussion of case studiesApplication state - elements without UIExposure: description, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesComponent based web frameworksExposure: description, explanation, expla		
Componentsexplanation, examples, discussion of case studies- composition vs inheritance Inferno js, https://github.com/infernojs/infernoof case studiesApplication state flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studiesElements properties and behaviors ormposition-Polymer, https://www.polymer-project.orgof case studiesApplication state elements without UI-7. Component based web frameworksExposure: description, examples, discussion of case studies- Polymer, https://www.polymer-project.org-Application state - elements without UI-7. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - composition properties and behaviors - composition Angular 2, https://angular.io/-Application state Angular 2, https://angular.io/-	5. Component based web frameworks	-
 properties, lifecycle, state, and events composition vs inheritance Inferno.js, https://github.com/infernojs/inferno Application state flux architecture Component based web frameworks Elements properties and behaviors composition Polymer, https://www.polymer-project.org Application state elements without UI Component based web frameworks Exposure: description, examples, discussion of case studies 		-
- composition vs inheritanceof case studies- Inferno.js, https://github.com/infernojs/infernoof case studiesApplication state flux architecture-6. Component based web frameworksExposure: description, explanation, examples, discussion of case studies- properties and behaviors - composition - Polymer, https://www.polymer-project.orgexamples, discussion of case studiesApplication state - elements without UI-7. Component based web frameworksExposure: description, explanation, examples, discussion of case studiesComponent based web frameworksExposure: description, explanation, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studiesApplication state - properties and behaviors - composition - Angular 2, https://angular.io/explanation, examples, discussion of case studiesApplication state-	±	-
 Inferno.js, https://github.com/infernojs/inferno Application state flux architecture Component based web frameworks Exposure:		· ·
Application state - flux architectureExposure: description, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, examples, discussion of case studies9. Polymer, https://www.polymer-project.orgApplication state - elements without UI7. Component based web frameworksExposure: description, explanation, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies9. Angular 2, https://angular.io/Application state	▲	of case studies
- flux architectureExposure: description, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, examples, discussion of case studies- properties and behaviors - composition - Polymer, https://www.polymer-project.orgexamples, discussion of case studiesApplication state - elements without UIExposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesApplication state - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies	- Inferno.js, https://github.com/infernojs/inferno	
- flux architectureExposure: description, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, examples, discussion of case studies- properties and behaviors - composition - Polymer, https://www.polymer-project.orgexamples, discussion of case studiesApplication state - elements without UIExposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesApplication state - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies		
6. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studies6. Component based web frameworksexplanation, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesApplication stateExposure: description, explanation, examples, discussion of case studiesApplication stateExposure: description, examples, discussion of case studies		
Lementsdescription, explanation, examples, discussion of case studies- properties and behaviorsexamples, discussion of case studies- Polymer, https://www.polymer-project.orgof case studiesApplication state - elements without UIExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studiesComponents and modules - properties and behaviors - composition - Angular 2, https://angular.io/examples, discussion of case studiesApplication stateImage: case studies		
Elementsexplanation,- properties and behaviorsexamples, discussion- compositionof case studies- Polymer, https://www.polymer-project.orgof case studiesApplication stateelements without UI- elements without UIExposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, examples, discussion of case studies- properties and behaviorsexamples, discussion of case studies- compositionof case studies- Angular 2, https://angular.io/of case studies	6. Component based web frameworks	-
 properties and behaviors composition Polymer, https://www.polymer-project.org Application state elements without UI Component based web frameworks Components and modules properties and behaviors composition Angular 2, https://angular.io/ Application state 		-
 composition Polymer, https://www.polymer-project.org Application state elements without UI Component based web frameworks Components and modules properties and behaviors composition Angular 2, https://angular.io/ Application state 		-
 Polymer, https://www.polymer-project.org Application state elements without UI 7. Component based web frameworks Components and modules properties and behaviors composition Angular 2, https://angular.io/ Application state 		
Application state - elements without UIExposure: description, explanation, explanation, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, examples, discussion of case studies6. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies	-	of case studies
- elements without UIExposure: description, explanation,7. Component based web frameworksExposure: description, explanation, explanation, of case studies- properties and behaviorsexamples, discussion of case studies- compositionof case studies- Angular 2, https://angular.io/	- Polymer, https://www.polymer-project.org	
- elements without UIExposure: description, explanation,7. Component based web frameworksExposure: description, explanation, explanation, of case studies- properties and behaviorsexamples, discussion of case studies- compositionof case studies- Angular 2, https://angular.io/	Application state	
7. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studies7. Components and modulesexposure: description, examples, discussion of case studies- properties and behaviorsexamples, discussion of case studies- Angular 2, https://angular.io/of case studies		
Components and modulesdescription, explanation, examples, discussion of case studies- properties and behaviorsexamples, discussion of case studies- compositionof case studies- Angular 2, https://angular.io/		.
Components and modulesexplanation,- properties and behaviorsexamples, discussion- compositionof case studies- Angular 2, https://angular.io/	7. Component based web frameworks	-
 properties and behaviors composition Angular 2, https://angular.io/ Application state examples, discussion of case studies 		-
 - composition - Angular 2, https://angular.io/ Application state of case studies 		-
- Angular 2, https://angular.io/ Application state		
Application state	-	of case studies
	- Angular 2, https://angular.io/	
- services		
	- services	

8. Creating a model-based framework for user interfaces IFML metamodel	Exposure: description, explanation, examples, discussion	
- domain model	of case studies	
- services, actions	of case studies	
- components, containers	Exposure	
9. Creating an IFML diagram editor	Exposure: description,	
componente contrinere	▲ ·	
- components, containers	explanation,	
- navigation flow	examples, discussion of case studies	
10. Creating a domain model diagram editor		
10. Creating a domain model diagram editor	Exposure:	
1 , ,.	description,	
- classes, properties, associations	explanation,	
	examples, discussion	
	of case studies	
11. Running and deploying components	Exposure:	
	description,	
- run component within the framework	explanation,	
- generate code and run components as standalone	examples, discussion	
apps	of case studies	
12. Component repository	Exposure:	
	description,	
- publish components	explanation,	
- reuse components	examples, discussion	
	of case studies	
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Creating a secured server for component	Dialogue, debate,	
repositories	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	Dialogue, debate,	
interfaces	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	Implement a system with	Project grading	100%				
activities	REST services, server side						
	notifications, and data						
	synchronization						
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