#### **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	Component Based Development

## 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 ofE2.7 Type ofOptional				
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					
Additional documentation (in libraries, on electronic platforms, field documentation)					
Preparation for seminars/labs, homework, papers, portfolios and essays					
Tutorship					
Evaluations					
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	<ul> <li>C 4.3 Identify models and methods adequate to real life problem solving</li> <li>C 2.1 Identify adequate software systems development methodologies</li> <li>C 1.1 Proper description of programming paradigms and language specific mechanisms, and identification of semantical an syntactical differences</li> </ul>
Tran svers al comp etenc ies	<ul> <li>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</li> <li>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</li> </ul>

# 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		

SESoftware Design Web frameworks for node based on - callback functions - generator functions - asyne functional reactive programming (FRP) - are functional reactive programming (FRP) - ecursion - recursion - recursion - recursion - recursion - recursion - map, reduce, filter - functional composition - intervent comp		
<ul> <li>callback functions         <ul> <li>agnerator functions</li> <li>agnerator functions</li> <li>agnerator functions</li> <li>agnerator functions</li> </ul> </li> <li>caltroposed functions</li> <li>agnerative programming (FRP)</li> <li>Exposure:             <ul></ul></li></ul>	SE/Software Design	
<ul> <li>generator functions</li> <li>asyne functions</li> <li>reactive extensions (rxjs)</li> <li>Functional reactive programming (FRP)</li> <li>pure functions, higher order functions</li> <li>explanation,</li> <li>explanat</li></ul>	Web frameworks for node based on	
<ul> <li>generator functions</li> <li>asyne functions</li> <li>reactive extensions (rxjs)</li> <li>Functional reactive programming (FRP)</li> <li>pure functions, higher order functions</li> <li>explanation,</li> <li>examples, discussion</li> <li>of case studies</li> </ul> - Cycle.js, https://cycle.js.org/ 4. Web frameworks based on FRP Exposure: <ul> <li>description,</li> <li>explanation,</li> <li>examples, discussion</li> <li>of case studies</li> </ul> - Revele.js, https://eycle.js.org/ 5. Component based web frameworks Exposure: <ul> <li>description,</li> <li>explanation,</li> <li>examples, discussion</li> </ul> of case studies - Revele.js, https://github.com/infernojs/inferno Application state <ul> <li>Porperties and behaviors</li> <li>elements</li> <li>elements<td>- callback functions</td><td></td></li></ul>	- callback functions	
- async functions - reactive extensions (rxjs) 2. Functional reactive programming (FRP) - pure functions, higher order functions - recursion - map, reduce, filter - functional composition 3. Web frameworks based on FRP description, - explanation, - examples, discussion Functional reactive programming - Cycle.js, https://cycle.js.org/ 4. Web frameworks based on FRP - Cycle.js, https://cycle.js.org/ 4. Web frameworks based on FRP - Cycle.js, https://cycle.js.org/ 4. Web frameworks based on FRP - Recycle.js, https://cycle.js.org/ 5. Component based on FRP - Recycle.js, https://cycle.js.org/ 5. Component based web frameworks - reoperties, https://cycle.js.org/ 5. Component based web frameworks - reoperties and behaviors - composition - properties and behaviors - composition - Polymer, https://www.polymer-project.org Application state - Polymer, https://www.polymer-project.org Application state - reoperties and behaviors - composition - composition - composition - properties and behaviors - composition - properties and behaviors - properties and behaviors		
- reactive extensions (xjs)       Exposure: description, explanation, examples, discussion of case studies         - pure functions, higher order functions recursion or map, reduce, filter       Exposure: description, explanation, examples, discussion of case studies         3.1 HCl/Programming Interactive Systems       Exposure: description, 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, examples, discussion of case studies         - Recycle.js, https://cycle.js.org/       Exposure: description, examples, discussion of case studies         - Recycle.js, https://cycle.js.org/       Exposure: description, examples, discussion of case studies         - Recycle.js, https://recycle.js.org/       Exposure: description, examples, discussion of case studies         - Recycle.js, https://recycle.js.org/       Exposure: description, examples, discussion of case studies         - Inferno.js, https://github.com/infernojs/inferno       Figure 4         - flux architecture       Exposure: description, explanation, examples, discussion of case studies         - Polymer, https://www.polymer-project.org       Exposure: description, examples, discussion of case studies         - Polymer, https://www.polymer-project.org	-	
2. Functional reactive programming (FRP)       Exposure:         - pure functions, higher order functions       explanation,         - recursion       explanation,         - map, reduce, filter       description,         - functional composition       explanation,         3. Web frameworks based on FRP       Exposure:         - functional reactive programming       of case studies         - Cycle.js, https://cycle.js.org/       Exposure:         4. Web frameworks based on FRP       Exposure:         - Gorporent based web frameworks       explanation,         Functional reactive programming       of case studies         - Cycle.js, https://cycle.js.org/       Exposure:         4. HCl/Programming Interactive Systems       explanation,         - Recycle.js, https://recycle.js.org/       Exposure:         - Component based web frameworks       Exposure:         - Component based web frameworks       explanation,         - Inferno.js, https://recycle.js.org/       explanation,         - Properties, lifecycle, state, and events       explanation,         - composition vs inheritance       explanation,         - Inferno.js, https://github.com/infernojs/inferno       description,         Application state       explanation,         - Properties and behaviors	-	
<ul> <li>pure functions, higher order functions</li> <li>pure functions, higher order functions</li> <li>recursion</li> <li>map, reduce, filter</li> <li>functional composition</li> <li>3. Web frameworks based on FRP</li> <li>geschift and the second secon</li></ul>		Exposure:
<ul> <li>- pure functions, higher order functions</li> <li>- recursion</li> <li>- functional composition</li> <li>- State of the state of</li></ul>	2. I unchonal reactive programming (I KI )	
- recursion - map, reduce, filter - of case studies - functional composition - map, reduce, filter - of case studies - functional composition - map, reduce, filter - of case studies - cycle.js, https://cycle.js.org/ - cycle.js, https://cycle.js.org/ cycle.js, https://cycle.js.org	- pure functions higher order functions	
- map, reduce, filterof case studies- functional compositionExposure: description, explanation, examples, discussion of case studies3.1 HCt/Programming Interactive SystemsExposure: description, examples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, examples, discussion of case studies- LittleExposure: description, examples, discussion of case studies- Recycle.js, https://cycle.js.org/Exposure: description, examples, discussion of case studies4.1 HCt/Programming Interactive SystemsExposure: description, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/Exposure: description, explanation, examples, discussion of case studiesComponent based web frameworksExposure: description, examples, discussion of case studiesComponents - properties, lifecycle, state, and events - compositionExposure: description, examples, discussion of case studiesApplication state - elements - properties and behaviors - properties and behaviors - properties and behaviors - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studiesApplication state - elements and modules - elements and modules - properties and behaviors - proper		-
- functional compositionExposure: description, examples, discussion of case studies3. Web frameworks based on FRPExposure: description, 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- Uter Programming - Recycle.js, https://recycle.js.org/Exposure: description, explanation, examples, discussion of case studies- Component based web frameworksExposure: description, explanation, examples, discussion of case studies- Components - properties, lifecycle, state, and events - composition vs inheritance - Inferno.js, https://github.com/infernojs/infernoExposure: description, examples, discussion of case studiesApplication state - elements - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, explanation, examples, discussion of case studiesApplication state - elements without UI - Component based web frameworksExposure: description, explanation, examples, discussion of case studiesApplication state - composition - properties and behaviors - composition - Reposure: description, explanation, examples, discussion of case studiesApplication state - properties and behaviors - composition - noposition - Angular 2, https://angular.io/Application state		
3. Web frameworks based on FRP       Exposure:         description,       explanation,         3.1 HCl/Programming 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. HCl/Programming Interactive Systems       explanation,         Functional reactive programming       of case studies         - Recycle.js, https://cycle.js.org/       examples, discussion         5. Component based web frameworks       Exposure:         composition vs inheritance       examples, discussion         - Inferno.js, https://github.com/infernojs/inferno       of case studies         Application state       exploanation,         - properties and behaviors       examples, discussion         - properties and behaviors       examples, discussion         - Polymer, https://www.polymer-project.org       examples, discussion         Application state       exploanation,         - elements without UI       examples, discussion         - Component based web frameworks       Exposure:         - Royner, https://www.pol	-	of case studies
3.1 HCt/Programming Interactive Systems       description, explanation, examples, discussion         Functional reactive programming       of case studies         - Cycle.js, https://cycle.js.org/       -         4. Web frameworks based on FRP       Exposure: description, explanation, examples, discussion         4. HCt/Programming Interactive Systems       explanation, examples, discussion         - Recycle.js, https://recycle.js.org/       -         5. Component based web frameworks       Exposure: description, explanation, examples, discussion         - orpoperties, lifecycle, state, and events       explanation, examples, discussion         - orpoperties, lifecycle, state, and events       explanation, examples, discussion         - Inferno.js, https://github.com/infernojs/inferno       of case studies         Application state       -         - properties and behaviors       explanation, examples, discussion         - composition       of case studies         - Polymer, https://www.polymer-project.org       Exposure: description, examples, discussion         Application state       -         - elements without UI       -         7. Component based web frameworks       Exposure: description, examples, discussion         Components and modules       explanation, explanation, explanation, explanation, examples, discussion         - properties and behaviors	4	Evenesium
3.1 HCl/Programming Interactive Systems       explanation, examples, discussion         Functional reactive programming       of case studies         - Cycle js, https://cycle.js.org/       Exposure:         description,       explanation,         4.1 HCl/Programming Interactive Systems       explanation,         explanation,       explanation,         Functional reactive programming       of case studies         Properties, https://recycle.js.org/       fease studies         5. Component based web frameworks       explanation,         explanation,       explanation,         composition vs inheritance       explanation,         - Infernojs, https://github.com/infernojs/inferno       of case studies         Application state       examples, discussion         - flux architecture       of case studies         Popperties and behaviors       examples, discussion         - properties and behaviors       examples, discussion         - properties and behaviors       examples, discussion         - Properties and behaviors       examples, discussion         - composition       of case studies         - Polymer, https://www.polymer-project.org       examples, discussion         Application state       elements         - elements without UI       explanation, <td>5. web frameworks based on FRP</td> <td>-</td>	5. web frameworks based on FRP	-
Functional reactive programmingexamples, discussion of case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies4.1 HCl/Programming Interactive Systemsexplanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/of case studies5. Component based web frameworksExposure: description, 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, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, examples, discussion of case studies9 properties and behaviors - composition - Recycle.gion - component based web frameworksExposure: description, explanati		1 '
Functional reactive programmingof case studies- Cycle.js, https://cycle.js.org/Exposure: description, explanation, examples, discussion of case studies4.1 HCl/Programming Interactive SystemsExposure: description, explanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/Exposure: description, explanation, examples, discussion of case studiesComponent based web frameworksExposure: description, explanation, examples, discussion of case studiesComponents - properties, https://github.com/infernojs/infernoexamples, discussion of case studiesApplication state - flux architectureExposure: description, examples, discussion of case studiesElements - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studiesApplication state - elements hutbus/ - properties and behaviors - composition - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, examples, discussion of case studies- Polymer, https://angular.jo/Exposure: description, examples, discussion of case studies- Angular 2, https://angular.jo/Application state- Angular 2, https://angular.jo/Application state	3.1 HCI/Programming Interactive Systems	
- Cycle.js, https://cycle.js.org/4. Web frameworks based on FRPExposure: description, examples, discussion of case studies4.1 HCl/Programming Interactive Systemsexplanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/of case studies5. Component based web frameworksExposure: description, explanation, examples, discussion of case studies- properties, lifecycle, state, and events - properties, https://github.com/infernojs/infernoexplanation, examples, discussion of case studiesApplication state - flux architectureExposure: description, explanation, explanation, examples, discussion of case studiesElements - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - composition - Angular 2, https://angular.jo/Exposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, <br< td=""><td></td><td>-</td></br<>		-
4. Web frameworks based on FRPExposure: description, explanation, examples, discussion4.1 HCl/Programming Interactive Systemsexplanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/Exposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, explanation, explanation, explanation, examples, discussion of case studiesApplication state - flux architectureExposure: description, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, explanation, explanation, examples, discussion of case studies- flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - 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 studies- Component based web frameworksExposure: description, examples, discussion of case studies- properties and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies <td>Functional reactive programming</td> <td>of case studies</td>	Functional reactive programming	of case studies
4. Web frameworks based on FRPExposure: description, explanation, examples, discussion4.1 HCl/Programming Interactive Systemsexplanation, examples, discussion of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/of case studiesFunctional reactive programming - Recycle.js, https://recycle.js.org/Exposure: description, examples, discussion of case studiesComponent based web frameworksExposure: description, explanation, explanation, explanation, examples, discussion of case studiesApplication state - flux architectureExposure: description, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, explanation, explanation, examples, discussion of case studies- flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - 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 studies- Component based web frameworksExposure: description, examples, discussion of case studies- properties and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies <td></td> <td></td>		
4.1 HCl/Programming Interactive Systemsdescription, explanation, examples, discussion of case studiesFunctional 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 - ormposition vs inheritance - Inferno.js, https://github.com/infernojs/infernoexamples, discussion of case studiesApplication state - flux architectureExposure: description, explanation, examples, discussion of case studiesElements - properties and behaviors - properties and behaviors - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studiesApplication state - elements without UIExposure: description, examples, discussion of case studies- Roponent based web frameworksExposure: description, examples, discussion of case studies- Polymer, https://www.polymer-project.orgexplanation, explana		
4.1 HCl/Programming Interactive Systems       explanation,         Functional reactive programming       examples, discussion         Functional reactive programming       examples, discussion         - Recycle.js, https://recycle.js.org/       explanation,         5. Component based web frameworks       Exposure:         description,       explanation,         - properties, lifecycle, state, and events       explanation,         - composition vs inheritance       explanation,         - Inferno.js, https://github.com/infernojs/inferno       of case studies         Application state       explanation,         - flux architecture       explanation,         6. Component based web frameworks       Exposure:         elements       explanation,         - properties and behaviors       explanation,         - Polymer, https://www.polymer-project.org       of case studies         Application state       -         - elements without UI       elements         - roporties and behaviors       explosure:         - elements without UI       explanation,         - properties and behaviors       explanation,         - properties and behaviors       explanation,         - properties and behaviors       explanation,         - properties and beh	4. Web frameworks based on FRP	
Functional reactive programming - Recycle.js, https://recycle.js.org/examples, discussion of case studies5. Component based web frameworksExposure: description, explanation, explanation, explanation, examples, discussion of case studiesComponents - properties, lifecycle, state, and events - composition vs inheritance - Inferno.js, https://github.com/infernojs/infernoExposure: description, examples, discussion of case studiesApplication state - flux architectureExposure: description, examples, discussion of case studies6. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studies9. Poperties and behaviors - composition - Properties and behaviors - Component based web frameworksExposure: description, examples, discussion of case studies9. Polymer, https://www.polymer-project.orgApplication state - elements without UI7. Component based web frameworksExposure: description, examples, discussion of case studies7. Component based web frameworksexplanation, examples, discussion of case studies9. Polymer, https://www.polymer-project.orgexplanation, examples, discussion of case studies9. Component based web frameworksexplanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies9. Application stateApplication state9. Application stateApplication state9. Application stateApplication state9. Application stateApplication state<		
Functional reactive programmingof case studies- Recycle.js, https://recycle.js.org/-5. Component based web frameworksExposure: description, explanation, 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 studies9. Component based web frameworksExposure: description, explanation, explanation, of case studies9. Properties and behaviors - composition - Polymer, https://www.polymer-project.org-Application state - elements without UI-7. Component based web frameworksExposure: description, examples, discussion of case studies- Roporties and behaviors - composition - properties and behaviors - elements without UIExposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesApplication state - elements and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studiesApplication state-	4.1 HCI/Programming Interactive Systems	1 · · ·
- Recycle.js, https://recycle.js.org/         5. Component based web frameworks       Exposure: description, explanation, explanation, examples, discussion of case studies         - Inferno.js, https://github.com/infernojs/inferno       of case studies         Application state - flux architecture       -         6. Component based web frameworks       Exposure: description, examples, discussion of case studies         Elements - properties and behaviors - composition - Polymer, https://www.polymer-project.org       Exposure: description, examples, discussion of case studies         Application state - elements without UI       Exposure: description, explanation, examples, discussion of case studies         7. Component based web frameworks       Exposure: description, explanation, explanation, examples, discussion of case studies         7. Components and modules - properties and behaviors - composition - composition - Angular 2, https://angular.io/       Exposure: description, examples, discussion of case studies		
5. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studiesApplication state - flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies9. Properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studies4Application state - flux architectureExposure: description, examples, discussion of case studies9. Polymer, https://www.polymer-project.orgFits and behaviors examples, discussion of case studies9. Component based web frameworksExposure: description, examples, discussion of case studies9. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies9. Application state	Functional reactive programming	of case studies
5. Component based web frameworksExposure: description, explanation, explanation, examples, discussion of case studiesApplication state - flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies9. Properties and behaviors - composition - Polymer, https://www.polymer-project.orgExposure: description, examples, discussion of case studies4Application state - flux architectureExposure: description, examples, discussion of case studies9. Polymer, https://www.polymer-project.orgFits and behaviors examples, discussion of case studies9. Component based web frameworksExposure: description, examples, discussion of case studies9. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies9. Application state	- Recycle.js, https://recycle.js.org/	
Componentsdescription, explanation, examples, discussion of case studies- properties, lifecycle, state, and eventsexamples, discussion of case studies- Inferno.js, https://github.com/infernojs/infernoof case studiesApplication state flux architecture-6. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors polymer, https://www.polymer-project.org-Application state - elements without UI-7. Component based web frameworksExposure: description, examples, discussion of case studies- ropperties and behaviors - composition - oroposition - oroposition - oroposition - oroposition - angular 2, https://angular.io/Exposure: description, examples, discussion of case studies- flux architecture elements without UI-7. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviors - properties and behaviors - composition - Angular 2, https://angular.io/-		Exposure:
Componentsexplanation, examples, discussion of case studies- roporties, lifecycle, state, and eventsexamples, discussion of case studies- Inferno.js, https://github.com/infernojs/infernoof case studiesApplication state flux architecture-6. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studiesApplication state - elements without UIExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, examples, discussion of case studies9. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies4pplication state-		-
<ul> <li>properties, lifecycle, state, and events</li> <li>composition vs inheritance</li> <li>Inferno.js, https://github.com/infernojs/inferno</li> <li>Application state</li> <li>flux architecture</li> <li>6. Component based web frameworks</li> <li>Elements</li> <li>properties and behaviors</li> <li>composition</li> <li>prolymer, https://www.polymer-project.org</li> <li>Application state</li> <li>elements without UI</li> <li>7. Component based web frameworks</li> <li>Exposure:</li> <li>description,</li> <li>examples, discussion</li> <li>of case studies</li> </ul>	Components	-
- composition vs inheritanceof case studies- Inferno.js, https://github.com/infernojs/infernoof case studiesApplication state flux architectureExposure: description, explanation, explanation, of case studies6. Component based web frameworksExposure: description, explanation, of case studies9. Properties and behaviors - composition - Polymer, https://www.polymer-project.orgexamples, discussion of case studiesApplication state - elements without UIExposure: description, explanation, explanation, explanation, of case studies7. Component based web frameworksExposure: description, 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/explanation examples, discussion of case studiesApplication stateApplication state	•	
- 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, examples, discussion of case studies Exposure: description, explanation, explanation, examples, discussion of case studies - elements without UI 7. Component based web frameworks - orpoperties and behaviors - properties and behaviors - composition - Angular 2, https://angular.io/		
Application state - flux architectureExposure: description, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, examples, discussion of case studies9. Polymer, https://www.polymer-project.orgPolymer, https://www.polymer-project.orgApplication state - elements without UIExposure: description, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, explanation, examples, discussion of case studies9. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies4pplication stateImage: discussion of case studies	<b>-</b>	
- flux architectureExposure: description, explanation, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, 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, examples, discussion of case studiesApplication state - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studiesApplication stateexplanation, examples, discussion of case studies	J., 11. 8-11. 11. 11. 11. 11. 11. 11. 11. 11. 11.	
- flux architectureExposure: description, explanation, explanation, examples, discussion of case studies6. Component based web frameworksExposure: description, 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, examples, discussion of case studiesApplication state - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studiesApplication stateexplanation, examples, discussion of case studies	Application state	
6. Component based web frameworksExposure: description, explanation, 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, of case studies7. Component based web frameworksExposure: description, explanation, explanation, explanation, of case studiesComponents and modules - properties and behaviors - composition - Angular 2, https://angular.io/explanation, examples, discussion of case studiesApplication state- description, examples, discussion of case studies		
Elementsdescription, explanation, examples, discussion of case studies- composition- composition of case studies- Polymer, https://www.polymer-project.org- composition of case studiesApplication state - elements without UI- component based web frameworks7. Component based web frameworksExposure: description, explanation, explanation, explanation, explanation, of case studiesComponents and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies		Exposure:
Elementsexplanation,- properties and behaviorsexamples, discussion- compositionof case studies- Polymer, https://www.polymer-project.orgof case studiesApplication stateelements without UI- elements without UIExposure: description, examples, discussion7. Component based web frameworksExposure: description, examples, discussion of case studies- properties and behaviorsexplanation, examples, discussion of case studies- compositionof case studies- Angular 2, https://angular.io/of case studies		-
<ul> <li>properties and behaviors</li> <li>composition</li> <li>Polymer, https://www.polymer-project.org</li> <li>Application state</li> <li>elements without UI</li> <li>Component based web frameworks</li> <li>Exposure: description, examples, discussion</li> <li>Exposure: description, explanation, explanation, examples, discussion</li> <li>of case studies</li> </ul>	Elements	
<ul> <li>- composition</li> <li>- Polymer, https://www.polymer-project.org</li> <li>- Application state</li> <li>- elements without UI</li> <li>7. Component based web frameworks</li> <li>- Components and modules</li> <li>- properties and behaviors</li> <li>- composition</li> <li>- composition</li> <li>- Angular 2, https://angular.io/</li> <li>Application state</li> </ul>		<b>1</b>
<ul> <li>Polymer, https://www.polymer-project.org</li> <li>Application state         <ul> <li>elements without UI</li> </ul> </li> <li>7. Component based web frameworks</li> <li>Components and modules         <ul> <li>properties and behaviors</li> <li>composition             <ul> <li>Angular 2, https://angular.io/</li> <li>Application state</li> <li>Langular 2, https://angular.io/</li> <li>Langular 2, https://angular.io/</li></ul></li></ul></li></ul>		-
Application state - elements without UIExposure: description, explanation, explanation, examples, discussion of case studies7. Component based web frameworksExposure: description, explanation, explanation, of case studies9. Components and modules - properties and behaviors - composition - Angular 2, https://angular.io/Exposure: description, examples, discussion of case studies	-	
- elements without UIExposure: description,7. Component based web frameworksExposure: description,Components and modulesexplanation,- properties and behaviorsexamples, discussion of case studies- compositionof case studies- Angular 2, https://angular.io/	r orymer, https://www.porymer-project.org	
- elements without UIExposure: description,7. Component based web frameworksExposure: description,Components and modulesexplanation,- 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, explanation, examples, discussion of case studies- properties and behaviors - composition - Angular 2, https://angular.io/examples, discussion of case studiesApplication state-		
Components and modulesdescription, explanation, examples, discussion of case studies- properties and behaviorsexamples, discussion of case studies- Angular 2, https://angular.io/		Exposure
Components and modulesexplanation, examples, discussion of case studies- properties and behaviorsexamples, discussion of case studies- Angular 2, https://angular.io/of case studies	7. Component based web frameworks	-
<ul> <li>properties and behaviors</li> <li>composition</li> <li>Angular 2, https://angular.io/</li> </ul> Application state <ul> <li>examples, discussion</li> <li>of case studies</li> </ul>	Components and module	-
<ul> <li>- composition</li> <li>- Angular 2, https://angular.io/</li> <li>Application state</li> <li>of case studies</li> </ul>	-	-
- Angular 2, https://angular.io/ Application state		
Application state	<b>▲</b>	of case studies
	- Angular 2, https://angular.io/	
- services		
	- services	

<ul><li>8. Creating a model-based framework for user interfaces</li><li>IFML metamodel</li></ul>	Exposure: description, explanation, examples, discussion	
- domain model	of case studies	
- services, actions		
- components, containers		
9. Creating an IFML diagram editor	Exposure:	
	description,	
- components, containers	explanation,	
- navigation flow	examples, discussion	
	of case studies	
10. Creating a domain model diagram editor	Exposure:	
	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:	
12. Component repository	description,	
- publish components	explanation,	
	examples, discussion	
- reuse components	of case studies	
	of case studies	
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Creating a secured server for component	Dialogue, debate,	Kennarks
repositories	case studies,	
repositories		
2 Creating a web ann based on EDD from works	examples, proofs	
2. Creating a web app based on FRP frameworks	Dialogue, debate,	
	case studies,	
2 Constinue and an interval and a second	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	Dialogua dahata	
or read component repository readines	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

- Р
- r
- 0
- f
- •
- \_
- d r
- .
- Α
- n
- c
- a
- Α
- n
- d
- u