

## SYLLABUS

### 1. Information regarding the programme

|                                     |  |
|-------------------------------------|--|
| 1.1 Higher education institution    | <b>Babeş Bolyai University</b>                     |
| 1.2 Faculty                         | <b>Faculty of Mathematics and Computer Science</b> |
| 1.3 Department                      | <b>Department of Computer Science</b>              |
| 1.4 Field of study                  | <b>Computer Science</b>                            |
| 1.5 Study cycle                     | <b>Master</b>                                      |
| 1.6 Study programme / Qualification | <b>Software Engineering</b>                        |

### 2. Information regarding the discipline

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

### 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   | <ul style="list-style-type: none"> <li>• Programming Fundamentals</li> </ul>  |
| 4.2. competencies | <ul style="list-style-type: none"> <li>• Good programming skills in at least one of the languages Java, C#</li> </ul> |

### 5. Conditions (if necessary)

|                                      |  |
|--------------------------------------|--|
| 5.1. for the course                  | <ul style="list-style-type: none"> <li>• Course hall with projector</li> </ul> |
| 5.2. for the seminar /lab activities | <ul style="list-style-type: none"> <li>• Laboratory with computers</li> </ul>  |

## 6. Specific competencies acquired

|                                  |   |
|----------------------------------|---|
| <b>Professional competencies</b> | <ul style="list-style-type: none"> <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 and syntactical differences</li> </ul>   |
| <b>Transversal competencies</b>  | <ul style="list-style-type: none"> <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)

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

## 8. Content

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

|   |   |  |
|---|---|--|
| <p>SE/Software Design</p> <p>Web frameworks for node based on</p> <ul style="list-style-type: none"> <li>- callback functions</li> <li>- generator functions</li> <li>- async functions</li> <li>- reactive extensions (rxjs)</li> </ul>  |   |  |
| <p>2. Functional reactive programming (FRP)</p> <ul style="list-style-type: none"> <li>- pure functions, higher order functions</li> <li>- recursion</li> <li>- map, reduce, filter</li> <li>- functional composition</li> </ul>  | <p>Exposure:<br/>description,<br/>explanation,<br/>examples, discussion<br/>of case studies</p> |  |
| <p>3. Web frameworks based on FRP</p> <p>3.1 HCI/Programming Interactive Systems</p> <p>Functional reactive programming</p> <ul style="list-style-type: none"> <li>- Cycle.js, <a href="https://cycle.js.org/">https://cycle.js.org/</a></li> </ul>   | <p>Exposure:<br/>description,<br/>explanation,<br/>examples, discussion<br/>of case studies</p> |  |
| <p>4. Web frameworks based on FRP</p> <p>4.1 HCI/Programming Interactive Systems</p> <p>Functional reactive programming</p> <ul style="list-style-type: none"> <li>- Recycle.js, <a href="https://recycle.js.org/">https://recycle.js.org/</a></li> </ul>   | <p>Exposure:<br/>description,<br/>explanation,<br/>examples, discussion<br/>of case studies</p> |  |
| <p>5. Component based web frameworks</p> <p>Components</p> <ul style="list-style-type: none"> <li>- properties, lifecycle, state, and events</li> <li>- composition vs inheritance</li> <li>- Inferno.js, <a href="https://github.com/infernojs/inferno">https://github.com/infernojs/inferno</a></li> </ul> <p>Application state</p> <ul style="list-style-type: none"> <li>- flux architecture</li> </ul> | <p>Exposure:<br/>description,<br/>explanation,<br/>examples, discussion<br/>of case studies</p> |  |
| <p>6. Component based web frameworks</p> <p>Elements</p> <ul style="list-style-type: none"> <li>- properties and behaviors</li> <li>- composition</li> <li>- Polymer, <a href="https://www.polymer-project.org">https://www.polymer-project.org</a></li> </ul> <p>Application state</p> <ul style="list-style-type: none"> <li>- elements without UI</li> </ul>   | <p>Exposure:<br/>description,<br/>explanation,<br/>examples, discussion<br/>of case studies</p> |  |
| <p>7. Component based web frameworks</p> <p>Components and modules</p> <ul style="list-style-type: none"> <li>- properties and behaviors</li> <li>- composition</li> <li>- Angular 2, <a href="https://angular.io/">https://angular.io/</a></li> </ul> <p>Application state</p> <ul style="list-style-type: none"> <li>- services</li> </ul>  | <p>Exposure:<br/>description,<br/>explanation,<br/>examples, discussion<br/>of case studies</p> |  |

|  |  |         |
|--|--|---------|
| 8. Creating a model-based framework for user interfaces<br><br>IFML metamodel<br>- domain model<br>- services, actions<br>- components, containers | Exposure:<br>description,<br>explanation,<br>examples, discussion<br>of case studies |         |
| 9. Creating an IFML diagram editor<br><br>- components, containers<br>- navigation flow  | Exposure:<br>description,<br>explanation,<br>examples, discussion<br>of case studies |         |
| 10. Creating a domain model diagram editor<br><br>- classes, properties, associations  | Exposure:<br>description,<br>explanation,<br>examples, discussion<br>of case studies |         |
| 11. Running and deploying components<br><br>- run component within the framework<br>- generate code and run components as standalone apps          | Exposure:<br>description,<br>explanation,<br>examples, discussion<br>of case studies |         |
| 12. Component repository<br><br>- publish components<br>- reuse components   | Exposure:<br>description,<br>explanation,<br>examples, discussion<br>of case studies |         |
|  |  |         |
| 8.2 Seminar / laboratory   | Teaching methods   | Remarks |
| 1. Creating a secured server for component repositories  | Dialogue, debate,<br>case studies,<br>examples, proofs                               |         |
| 2. Creating a web app based on FRP frameworks  | Dialogue, debate,<br>case studies,<br>examples, proofs                               |         |
| 3. Creating a web app based on web components  | Dialogue, debate,<br>case studies,<br>examples, proofs                               |         |
| 4. Creating a model-based framework for user interfaces  | Dialogue, debate,<br>case studies,<br>examples, proofs                               |         |
| 5. Add diagram editors   | Dialogue, debate,<br>case studies,<br>examples, proofs                               |         |
| 6. Add component repository features   | Dialogue, debate,<br>case studies,<br>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 Curricula 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   |  |                         |                             |
| <ul style="list-style-type: none"> <li>➤ 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.</li> <li>➤ No more than 3 absences are allowed for the seminar/lab activities</li> </ul> |  |                         |                             |

Date

20.04.18

Signature of course coordinator

**Lect. dr. Ioan Lazar**

Signature of seminar coordinator

**Lect. dr. Ioan Lazar**

Date of approval

Signature of the head of department

**P  
r  
o  
f  
.  
  
d  
r  
.  
  
A  
n  
c  
a  
  
A  
n  
d**