

## SYLLABUS

### 1. Information regarding the programme

1.1 Higher education institution	<b>“Babes-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>Informatics(Computer Science)</b>
1.5 Study cycle	<b>Bachelor</b>
1.6 Study programme / Qualification	<b>Computer Science</b>

### 2. Information regarding the discipline

2.1 Name of the discipline	<b>Multi-paradigm programming languages</b>						
2.2 Course coordinator	<b>Assoc.Prof.PhD. Niculescu Virginia</b>						
2.3 Seminar coordinator	<b>Assoc.Prof.PhD. Niculescu Virginia</b>						
2.4. Year of study	<b>3</b>	2.5 Semester	<b>6</b>	2.6. Type of evaluation	<b>C.</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 sem. +1pr.
3.4 Total hours in the curriculum	48	Of which: 3.5 course	24	3.6 seminar/laboratory	24
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					15
Additional documentation (in libraries, on electronic platforms, field documentation)					27
Preparation for seminars/labs, homework, papers, portfolios and essays					35
Tutorship					4
Evaluations					10
Other activities: .....					-
3.7 Total individual study hours					77
3.8 Total hours per semester					125
3.9 Number of ECTS credits					5

### 4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> <li>• Fundamentals of Programming</li> <li>• Object-Oriented Programming, Functional Programming</li> <li>• Java, C++ programming</li> </ul>
4.2. competencies	<ul style="list-style-type: none"> <li>• Programming skills and basic abilities for dealing with abstractions</li> </ul>

## 5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> <li>projector</li> </ul>
5.2. for the seminar	<ul style="list-style-type: none"> <li>projector</li> </ul>

## 6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> <li>Each student has to prove that (s)he acquired an acceptable level of knowledge and understanding of the subject, that (s)he is capable of stating this knowledge in a coherent form, that (s)he has correct habits of analysis, design of problems related to workflow systems.</li> </ul>
Transversal competencies	<ul style="list-style-type: none"> <li>Good communication skills; concise and precise description, both oral and written.</li> <li>Independent work capabilities; able to fulfill different roles.</li> </ul>

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

7.1 General objective of the discipline	<ul style="list-style-type: none"> <li>Interpretation of mathematical and programming models.</li> <li>Enhance the students understanding of programming paradigms</li> <li>To emphasize the relation between theory of programming paradigms with their concrete applications in languages</li> </ul>
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> <li>Deepen the knowledge of programming in Java, C#, C++.</li> <li>Learn new programming languages.</li> <li>Introduction in Scala programming and other multiparadigm languages (Kotlin, Go, Ruby, Julia...).</li> </ul>

## 8. Content

8.1 Course	Teaching methods	Remarks
1. Programming paradigm overview	Exposure: description, explanation, examples, discussion of case studies	
2. Basics of Scala Programming	Exposure: description, explanation, examples, discussion of case studies	
3. Object oriented programming in Scala	Exposure: description, explanation, examples, discussion of case studies	
4. Functional programming in Scala	Exposure: description, explanation, examples, discussion of case studies	
5. Concurrency in Scala	Exposure: description, explanation, examples, discussion of case studies	

6. Parallel programming paradigms in Scala	Exposure: description, explanation, examples, discussion of case studies	
7. Introduction in Scala coroutines	Exposure: description, explanation, examples, discussion of case studies	
8. Actor programming with Scala (Akka)	Exposure: description, explanation, examples, discussion of case studies	
9. Metaprogramming	Exposure: description, explanation, examples, discussion of case studies	
10. Dataflow programming	Exposure: description, explanation, examples, discussion of case studies	
11. Essay and project presentation	Exposure: description, explanation, examples, discussion of case studies	
12. Essay and project presentation	Exposure: description, explanation, examples, discussion of case studies	

### Bibliography

1. Michael Scott. Programming Language Pragmatics. 4<sup>th</sup> ed. Morgan Kaufmann, 2015
2. Martin Odersky, Lex Spoon, Bill Venners. Programming in Scala: A Comprehensive Step-by-Step Guide, 2nd Edition
3. Paul Chiusano and Runar Bjarnason Functional Programming in Scala, Mannon, 2014.
4. Bjarne Stroustrup: The C++ Programming Language Special Edition, Addison-Wesley, 2000
5. Andrei Alexandrescu. Modern C++ Design: Generic Programming and Design Patterns Applied Addison-Wesley Professional.2001
6. Georgy Pashkov Multi-Paradigm Programming with Modern C++. Packt Publishing.2020
7. David Vandevoorde, Douglas Gregor, Nicolai M. Josuttis. C++ Templates: The Complete Guide, 2nd Edition. Addison-Wesley Professional. 2017 Chapter: C++ Metaprogramming.
8. C.D. Marlin. Coroutines. A Programming Methodology, a Language Design and an Implementation. Springer-Verlag Berlin Heidelberg

8.2 Laborator	Teaching methods	Remarks
1. Scala simple example	Explanation, dialogue, case studies	The seminar is structured as 2 hours classes every second week
2. Scala programming assignment 1	Dialogue, debate, case studies, examples, proofs	

3. Scala programming assignment 2	Dialogue, debate, case studies, examples, proofs	
4. Scala programming assignment 3	Dialogue, debate, explanation, examples	
5. Student presentations	Dialogue, debate, explanation, examples	
6. Student presentations	Dialogue, debate, explanation, examples	
<b>Bibliography</b> <ol style="list-style-type: none"> <li>1. Michel Schinz and Philipp Haller. A Scala Tutorial for Java Programmers. <a href="https://docs.scala-lang.org/tutorials/scala-for-java-programmers.html">https://docs.scala-lang.org/tutorials/scala-for-java-programmers.html</a></li> <li>2. Scala Tutorial. <a href="https://www.tutorialspoint.com/scala/index.htm">https://www.tutorialspoint.com/scala/index.htm</a></li> </ol>		

**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 content of the course is considered of interest by the software companies

**10. Evaluation**

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Laboratory assignments -	Presentations, Questions/analysis	40%
10.5 Seminar	- Essay that presents a multiparadigm language and a mini project done in the chosen language.	-Presentation -discussion, critical analysis	60%
10.6 Minimum performance standards			
➤ At least grade 5 (from a scale of 1 to 10) for the final grade.			

Date  
.....

Signature of course coordinator  
.....Niculescu Virginia.....

Signature of seminar coordinator  
.....Niculescu Virginia

Date of approval  
.....

Signature of the head of department  
.....