Module title: Programming Paradigms

<table>
<thead>
<tr>
<th>Module ID</th>
<th>Workload</th>
<th>Credits</th>
<th>Semester</th>
<th>Frequency of Offering</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI11</td>
<td>150 h</td>
<td>5</td>
<td>1</td>
<td>yearly</td>
<td>1 semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workload</th>
<th>Attendance</th>
<th>Preparation and Follow-Up</th>
<th>Private Study</th>
<th>Preparation for Exam and Exam</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>30 h / 2 SWS</td>
<td>15 h</td>
<td>45 h</td>
<td>10 h</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>15 h / 1 SWS</td>
<td>45 h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>45 h / 3 SWS</td>
<td>60 h</td>
<td>45 h</td>
<td>10 h</td>
<td>150 h</td>
</tr>
</tbody>
</table>

1 Scheduled Group Size: S: 15 students, P: 15 students

2 Subject Knowledge / Skills

The objective target of this module is the acquisition of basic knowledge of the most prominent current programming paradigms. In addition to substantial basic knowledge in imperative and object oriented programming, students will gain a good understanding in the areas of functional and logic programming.

The students are able to compare different approaches to programming and are able to identify and name commonalities and differences between programming languages. They know and understand the conditions under which a specific programming paradigm is applicable and are able to specify the limits of at least four different programming paradigms. They have gained insight into the historical development of programming languages and are able to recognise these concepts in new programming languages.

3 Content / Syllabus

Introduction to programming and programming languages
- A short history of programming languages
- The concept of infinity
- A short introduction to the theory of computation

Imperative programming
- Modular programming with functions and procedures
- Divide and Conquer as a basic programming concept
- Example languages: Python

Object oriented programming
- Basic concepts: classes, objects, inheritance, polymorphism
- Programming with interfaces
- Aggregation and composition
- Design patterns
- Example languages: C# and Java

Functional programming
- Mathematical Notation: Lambda calculus and currying
- Higher order functions
- Comprehension and memoization
- Example languages: F# and Scheme
Logic programming
  Logic notation using Horn clauses
  Unification as a basic reasoning mechanism

4 Teaching Format
Seminar accompanied by practical work in a laboratory environment

5 Prerequisites
None

6 Recommended Qualifications for the Participation
None

7 Assessment
Written exam

8 Prerequisites for Granting ECTS Credits
Exam passed

9 Usage of this Module in Other Degree Courses
Master Applied Computer Science

10 Contribution to Final Score
5,56 %

11 Convenor
Professor of Digital Media Computing and Web Technologies

12 Language of Instruction
English

13 Reading List
- Bishop, Judith. C# 3.0 Design Patterns, O'Reilly Media, 2008.