Scientific C++ Programming (Advanced)

<table>
<thead>
<tr>
<th>Module-No./Abbreviation</th>
<th>Credits</th>
<th>Workload</th>
<th>Term</th>
<th>Frequency</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE-W10/SCP A</td>
<td>3 CP</td>
<td>90 h</td>
<td>2nd Sem.</td>
<td>Summer term</td>
<td>1 Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scientific C++ Programming (Advanced)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contact hours</th>
<th>Self-Study</th>
<th>Group Size:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 SWS (30 h)</td>
<td>60 h</td>
<td>No Restrictions</td>
</tr>
</tbody>
</table>

**Prerequisites**

- 

**Learning goals / Competences:**

After successfully completing the module, the students

- are familiar with advanced programming concepts and constructs in C++,
- are able to design and develop modern C++ applications using latest language features,
- can review and contribute to advanced C++ projects.

**Content**

The lecture addresses advanced topics in C++ programming. Object-oriented programming concepts such as classes, inheritance and polymorphism as well as generic programming concepts such as templates are introduced. The standard template library (STL) and selected functionalities from C++14 and above are surveyed. Best practices as well as the organization and development of advanced C++ projects are discussed.

In hands-on sessions, programming exercises are used to discuss and illustrate the presented content.

**Teaching methods / Language**

Block course (equiv. to 2 SWS) / English

**Mode of assessment**

Written examination (120 min., 100%)

**Requirement for the award of credit points**

Passed final module examination

**Module applicability**

- 

**Weight of the mark for the final score**

- 

**Module coordinator and lecturer(s)**

Prof. Dr. A. Vogel, Assistants

**Further information**