Degree: M.Sc. in Physics (PO von 2014)

Modules:

- physics70a Elective Advanced Lectures: Experimental Physics
- physics70b Elective Advanced Lectures: Applied Physics

Course: Programming in Physics and Astronomy with C++ or Python (E/A)

Course No.: physics718

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Language</th>
<th>Teaching hours</th>
<th>CP</th>
<th>Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elective</td>
<td>Lecture with exercises</td>
<td>English</td>
<td>2+1</td>
<td>4</td>
<td>ST</td>
</tr>
</tbody>
</table>

Requirements for Participation:

Preparation:
Basic knowledge of programming and knowledge of simple C/C++ or Python constructs.

Form of Testing and Examination:
C/C++ part: Requirements for the examination (written or oral): successful work with the exercises.
Python part: Requirements for examination: successful implementation of the scientific projects in Python during the semester.

Length of Course:
1 semester

Aims of the Course:
C++ part: In-depth understanding of C++ and its applications in particle physics. Discussion of advanced features of C++ using examples from High Energy Physics. The course is intended for students with some background in C++ or for advanced students who wish to apply C++ in their graduate research.

Python part: Effective and flexible program solving with the easy-to-learn, high level programming language Python. The course addresses master and PhD students with prior Python-programming knowledge as taught in the bachelor course physics131.

Contents of the Course:
C++ part: - Basic ingredients of C++, - Object orientation: classes, inheritance, polymorphism, - How to solve physics problems with C++, - Standard Template Library, - C++ in data analysis, example: the ROOT library, - C++ and large scale calculations, - How to write and maintain complex programs, - Parallel computing and the Grid, - Debugging and profiling

Python part: - In-depth introduction to Python based on prior programming experience, - Introduction to numpy arrays (primary Python data structure for scientific computing), - Introduction to scientific-Python modules (scipy, astropy), - Interactive work / development with Python (ipython), - Web interaction with Python (jupyter notebooks, web and database queries), - Plotting with Python (the matplotlib module)

Recommended Literature:
Lippman, Lajoie, Moo: C++ Primer, Addison-Wesley 2000.
Deitel and Deitel, C++ how to program, Prentice Hall 2007.

- The course is given in the summer term and alternates between C++ and Python
- The course can only be taken once for credit points.

January 2018