

Modulverantwortliche/r	Prof. Dr. Lena Oden				
	<table border="0"> <tr> <td>Dauer des Moduls ein Semester</td> <td>ECTS 10</td> <td>Workload 300 Stunden</td> <td>Häufigkeit in jedem Wintersemester</td> </tr> </table>	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Wintersemester
Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Wintersemester		
Lehrveranstaltung(en)	Fachpraktikum Scientific Programming in Python				
Detaillierter Zeitaufwand	<p>Präsenzphase am Ende des Semesters: 25 Stunden Vorbereitung und Einarbeitung: 50 Stunden Bearbeitung der vorgegebenen Programmieraufgaben: 100 Stunden Bearbeitung eines Projektes im Team: 100 Stunden Dokumentation der Ergebnisse: 25 Stunden</p>				
Qualifikationsziele	<p>Nach dem Abschluss des Praktikums sind die Studierenden in der Lage, mathematische, vor allem numerische Probleme, effizient mit Python zu lösen. Dies umfasst das Arbeiten mit Matrizen und Vektoren, das Lösen von Gleichungssystemen, Datenanalyse und die ansprechende Visualisierung der Ergebnisse. Sie kennen die Werkzeuge, wie diese Probleme auf moderne Hardware effizient gelöst werden können, und sind in der Lage, die Lösung auch einfach zu optimieren und zu parallelisieren. Sie kennen die wichtigsten Pakete zum effizienten arbeiten mit Python, wie numba, scipy und Pandas. Sie haben gelernt, umfangreiche Programmieraufgaben in Team zu bearbeiten.</p>				
Inhalte	<p>Unter Scientific Computing versteht man die Umsetzung numerischer Algorithmen in eine Programmiersprache, um wissenschaftliche Probleme zu lösen. Für viele wissenschaftliche Algorithmen sind Geschwindigkeit und auch der Speicherverbrauch sehr wichtig. Reines Python hat jedoch einen hohen Speicherverbrauch und ist vergleichsweise - langsam, es bietet jedoch viele nützliche Erweiterungen an, mit deren Hilfe sich numerische Probleme effektiv lösen lassen.</p> <p>In dem Praktikum soll die effektive Nutzung dieser Pakete für wissenschaftliches Arbeiten erlernt werden. Dabei wird es zunächst schwerpunktmäßig um Numpy, ScipY, Matplotlib und Pandas gehen. Dabei wird es um die Lösung numerischer Probleme, die Analyse großer Datenmengen und das Lösen von Simulations-Aufgaben gehen. Ein weitere wichtiger Teil des Praktikums ist die ansprechende Visualisierung der Ergebnisse.</p> <p>Im zweiten Teil des Praktikums geht es dann um Numba und mpi4py, welche eine effiziente Parallelisierung der Algorithmen ermöglichen.</p>				
Inhaltliche Voraussetzung	<p>Grundlegende Programierkenntnisse Grundlegende Kenntnisse in Numerik und Analysis</p>				
Lehr- und Betreuungsformen	<p>Studientag/e Betreuung und Beratung durch Lehrende Zusatzmaterial</p>				
Anmerkung	<p>Für die Teilnahme an dieser Veranstaltung ist ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: https://webregis.fernuni-hagen.de.</p>				
Formale Voraussetzung	<p>Studieneingangsphase ist abgeschlossen, die Module 63081 "Grundpraktikum Programmierung", 63912 "Grundlagen der Theoretischen Informatik" und 63012 "Softwaresysteme" sind bestanden</p>				
Verwendung des Moduls	<p>B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik</p>				

Prüfungsformen

Prüfung

Stellenwert
der Note 1/16

Art der Prüfungsleistung

benotete Praktikumsteilnahme
(Ausarbeitung und Vortrag)

Voraussetzung

keine