

# Modulhandbuch

## M. Sc. Computer Science

Fachbereich Informatik  
Technische Universität Darmstadt



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



Informatik



## **Modulhandbuch M. Sc. Computer Science**

Technische Universität Darmstadt

Fachbereich Informatik

Hochschulstr. 10

64289 Darmstadt

### **Redaktion**

Dipl.-Inform. Tim Neubacher

Jasmin Boghrat, M.A.

**Stand: 11.05.2023**

# Inhaltsverzeichnis

## Wahlbereiche

Wahlbereich Software und Hardware	5
Wahlbereich Theorie	51

## Vertiefungen

### Data Science and Engineering

#### Wahlbereich Fachprüfungen

Wahlbereich Data Science Applications	60
Wahlbereich Data Systems Engineering	75
Wahlbereich Foundations of Data Science	93

#### Wahlbereich Studienbegleitende Leistungen

Praktika, Projektpraktika und ähnliche Veranstaltungen	105
Seminare	137
Praktikum in der Lehre	164

### Distributed Computing

#### Wahlbereich Fachprüfungen

Wahlbereich Computer Networks and Distributed Systems	175
Wahlbereich Data-Intensive Systems and heterogeneous Hardware	189
Wahlbereich System Modelling and Engineering	198

#### Wahlbereich Studienbegleitende Leistungen

Praktika, Projektpraktika und ähnliche Veranstaltungen	207
Seminare	236
Praktikum in der Lehre	259

---

## **Visual Computing**

### **Wahlbereich Fachprüfungen**

Wahlbereich Computer Graphics	270
Wahlbereich Computer Vision und Machine Learning	284
Wahlbereich Integrated Methods of Graphics and Vision	311

### **Wahlbereich Studienbegleitende Leistungen**

Praktika, Projektpraktika und ähnliche Veranstaltungen	328
Seminare	359
Praktikum in der Lehre	386

<b>Masterarbeit</b>	<b>399</b>
---------------------	------------

**Modulhandbuch**  
**M. Sc. Computer Science**

**Wahlbereich Software und Hardware**  
**(Praktische, technische und angewandte Informatik)**

## Modulbeschreibung

<b>Modulname</b> Graphische Datenverarbeitung I					
<b>Modul Nr.</b> 20-00-0040	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0040-iv	Graphische Datenverarbeitung I	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Einführung in die Grundlagen der Computergraphik, insb. Ein- u. Ausgabegeräte, Rendering Pipeline am Beispiel von OpenGL, räumliche Datenstrukturen, Beleuchtungsmodelle, Ray Tracing, aktuelle Entwicklungen in der Computergraphik				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch dieser Veranstaltung sind Studierende in der Lage alle Komponenten der Graphikpipeline zu verstehen und dadurch variable Bestandteile (Vertex-Shader, Fragment-Shader, etc.) anzupassen. Sie können Objekte im 3D-Raum anordnen, verändern und effektiv speichern, sowie die Kamera und die Perspektive entsprechend wählen und verschiedene Shading-Techniken und Beleuchtungsmodelle nutzen, um alle Schritte auf dem Weg zum dargestellten 2D-Bild anzupassen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: <ul style="list-style-type: none"><li>• Programmierkenntnisse</li><li>• Kenntnisse über grundlegende Algorithmen und Datenstrukturen</li><li>• Kenntnisse im Bereich Lineare Algebra</li><li>• Kenntnisse im Bereich Analysis</li><li>• Inhalte der Vorlesung „Visual Computing“ oder einer vergleichbaren Veranstaltung</li></ul>				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0040-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0040-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M.Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <ul style="list-style-type: none"> <li>• Real-Time Rendering: Tomas Akenine-Möller, Eric Haines, Naty Hoffman A.K. Peters Ltd., 3rd edition, ISBN 987-1-56881-424-7</li> <li>• Fundamentals of Computer Graphics: Peter Shirley, Steve Marschner, third edition, ISBN 979-1-56881-469-8</li> <li>• Weitere aktuelle Literaturhinweise werden in der Veranstaltung gegeben.</li> </ul>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Graphische Datenverarbeitung II					
<b>Modul Nr.</b> 20-00-0041	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0041-iv	Graphische Datenverarbeitung II	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Grundlagen der verschiedenen Objekt- und Oberflächen-Repräsentationen in der graphischen Datenverarbeitung. Kurven und Oberflächen (Polynome, Splines, RBF) Interpolation und Approximation, Displaytechniken, Algorithmen: de Casteljau, de Boor, Oslo, etc. Volumen und implizite Oberflächen. Visualisierungstechniken, Iso-Surfaces, MLS, Oberflächen-Rendering, Marching-Cubes. Polygonnetze. Netz Kompression , Netz-Vereinfachung, Multiskalen Darstellung, Subdivision. Punktwolken: Renderingtechniken, Oberflächen-Rekonstruktion, Voronoi-Diagramme und Delaunay-Triangulierung.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch dieser Veranstaltung sind Studierende in der Lage mit diversen Objekt- und Oberflächen-Repräsentationen umzugehen, das heißt diese zu verwenden, anzupassen, anzuzeigen (rendern) und effektiv zu speichern. Dazu gehören mathematisch polynomiale Repräsentationen, Iso-oberflächen, volumen Darstellungen, implizite Oberflächen, Polygonnetze, Subdivision-Kontrollnetze und Punktwolken.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Der vorherige Besuch von „Algorithmen und Datenstrukturen“ und „Graphische Datenverarbeitung I“ oder vergleichbaren Veranstaltungen  Kenntnisse über Grundlagen aus der Höheren Mathematik  Programmierkenntnisse in C / C++				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0041-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				



	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0041-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <ul style="list-style-type: none"> <li>• Real-Time Rendering: Tomas Akenine-Möller, Eric Haines, Naty Hoffman A.K. Peters Ltd., 3rd edition, ISBN 987-1-56881-424-7</li> <li>• Weitere aktuelle Literaturhinweise werden in der Veranstaltung gegeben.</li> </ul>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Bildverarbeitung					
<b>Modul Nr.</b> 20-00-0155	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0155-iv	Bildverarbeitung	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> Überblick über die Grundlagen der Bildverarbeitung: - Bildeigenschaften - Bildtransformationen - einfache und komplexere Filterung - Bildkompression, - Segmentierung - Klassifikation				
3	<b>Qualifikationsziele / Lernergebnisse</b> Noch erfolgreichem Besuch der Veranstaltung haben die Studierenden einen Überblick über die Funktionsweise und die Möglichkeiten der modernen Bildverarbeitung. Studierende sind dazu in der Lage, einfache bis mittlere Bildverarbeitungsaufgaben selbständig zu lösen.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0155-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0155-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> <ul style="list-style-type: none"> <li>• Gonzalez, R.C., Woods, R.E., "Digital Image Processing", Addison- Wesley Publishing Company, 1992</li> <li>• Haberaecker, P., "Praxis der Digitalen Bildverarbeitung und Mustererkennung", Carl Hanser Verlag, 1995</li> <li>• Jaehne, B., "Digitale Bildverarbeitung", Springer Verlag, 1997</li> </ul>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Virtuelle und Erweiterte Realität					
<b>Modul Nr.</b> 20-00-0160	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0160-iv	Virtuelle und Erweiterte Realität	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Im Rahmen dieser Lehrveranstaltung werden zuerst die Grundlagen, Begriffsbildungen und Referenzmodelle zur Einordnung der Thematik im Rahmen der Computer-Graphik/Computer-Vision aufgezeigt. Aufbauend darauf werden die besonderen Technologien, Algorithmen und Standards der Augmented Reality (AR) und der Virtual Reality (VR) behandelt. Dazu gehören: <ul style="list-style-type: none"><li>• Datenschnittstellen (Standards, Vorverarbeitung, Systeme, etc.)</li><li>• Interaktionstechniken (z.B. Interaktion mit Hilfe von Rangekameras)</li><li>• Darstellungsverfahren (z.B. Echtzeit-Rendering)</li><li>• Web-basierte VR/AR</li><li>• Computer-Vision-basiertes Tracking für Augmented-Reality</li><li>• Augmented Reality mit Rangekamera-Technologien</li><li>• Augmented Reality auf Smartphonesystemen</li></ul> Schließlich werden diese Techniken an Beispielen aktueller Forschungsarbeiten aus den Bereichen „AR/VR-Wartungsunterstützung“ und „AR/VR-gestützte Präsentation von Kulturgütern“ dokumentiert.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung die Anforderungen und Problematiken von Virtual/Augmented Reality und sie wissen, für welche Problemstellungen diese Technologien eingesetzt werden können. Sie kennen die Standards, mit deren Hilfe VR/AR-Anwendungen spezifiziert werden, insb. wissen die Studierenden, welche Computer-Vision-Technologien eingesetzt werden können, um in verschiedenen Umgebungen die Kamerapose stabil zu tracken.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen der Graphischen Datenverarbeitung (GDV)				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0160-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0160-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Dörner, R., Broll, W., Grimm, P., Jung, B. Virtual und Augmented Reality (VR / AR)
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Algorithmen für Hardware-Entwurfswerkzeuge					
<b>Modul Nr.</b> 20-00-0183	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0183-v1	Algorithmen für Hardware-Entwurfswerkzeuge	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> - Das VLSI-Entwurfsproblem - Grundlegende Graphenrepräsentationen und -algorithmen - Darstellung von hierarchischen Schaltungen - Realisierungstechnologien für integrierte Schaltungen - Layout-Kompaktierung - Timing-Analyse - Heuristische Optimierungsverfahren - Platzierungsprobleme, -verfahren und -kostenfunktionen - Exakte Optimierungsverfahren - Partitionierung mit Anwendung in der Platzierung - Floorplanningprobleme, -repräsentationen und -verfahren - Verdrahtungsprobleme, -verfahren und -kostenfunktionen				
3	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung verschiedene Technologien für die Realisierung von integrierten Schaltungen. Sie können aus den verschiedenen Technologien die Anforderungen an Automatisierungswerkzeuge für verschiedene Teilaufgaben des Entwurfs- und Realisierungsprozesses herleiten. Sie sind vertraut mit der Modellierung technologischer Probleme durch formale Konzepte wie Graphen, Gleichungssysteme etc. Sie verstehen grundlegende Verfahren zur Lösung auch von harten Problemen und können aufbauend auf Erfahrungen mit verschiedenen Basisalgorithmen neue bzw. verfeinerte Implementierungen zur Erledigung der Entwurfsaufgaben entwickeln.				
4	<b>Voraussetzung für die Teilnahme</b>				

	Empfohlen: Empfohlen wird der erfolgreiche Besuch der Veranstaltungen “Digitaltechnik” sowie “Algorithmen und Datenstrukturen” und “Funktionale und objektorientierte Programmierung” oder vergleichbarer Veranstaltungen.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0183-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0183-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein: Gerez: Algorithms for VLSI Design Automation Wang/Chang/Cheng: Electronic Design Automation
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Informationsvisualisierung und Visual Analytics					
<b>Modul Nr.</b> 20-00-0294	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0294-iv	Informationsvisualisierung und Visual Analytics	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Diese Vorlesung wird eine detaillierte Einführung in die Informationsvisualisierung geben, um sich dann intensiv den wissenschaftlichen Fragestellungen und praxisnahen Anwendungsszenarien von Visual Analytics zu widmen. •Überblick der Informationsvisualisierung und Visual Analytics (Definitionen, Modelle, Historie) •Datenpräsentierung und Datentransformation •Abbildung von Daten auf visuelle Strukturen • Visuelle Repräsentierungen und Interaktion fuer bivariate, multivariate Daten, Zeitreihen, Graphen und Geographische Daten • Grundlagen von Data Mining •Grundlagen von Visual Analytics: - Analytische Beweisführung - Data Mining • Evaluation von Visual Analytics Systemen  Anwendungsgebiete: Medizin, Biologie, Finanzen und Wirtschaft, Meteorologie, Rettungsdienst,....				
3	<b>Qualifikationsziele / Lernergebnisse</b> Studierende können nach erfolgreichem Besuch der Veranstaltung  •Informationsvisualisierungsmethoden für verschiedene Datentypen benutzen  •interactive Visualisierungssysteme für Daten aus verschiedenen Anwendungsgebieten designen  •Visualisierung und automatische Datenverarbeitung kombinieren um Big Data Probleme zu lösen				



	<ul style="list-style-type: none"> <li>• Wissen über Hauptcharakteristika menschlicher visuellen Wahrnehmung in Informationsvisualisierung und Visual Analytics anwenden</li> <li>• geeignete Evaluationsmethode für spezifische Situationen und Szenarien auswählen</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Interesse an Methoden der Computergrafik und Visualisierung
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0294-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0294-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird in der Vorlesung bekanntgegeben. Beispiele für verwendete Literatur könnten sein: C. Ware: Information Visualization: Perception for Design Ellis et al: Mastering the Information Age
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Medizinische Bildverarbeitung					
<b>Modul Nr.</b> 20-00-0379	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0379-v1	Medizinische Bildverarbeitung	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> Die Vorlesung gliedert sich in zwei Teile. In der ersten Hälfte der Vorlesung wird die Funktionsweise von Geräten, welche medizinische Bilder liefern (CT, MRI, PET, SPECT, Ultraschall), erklärt. In der zweiten Hälfte werden verschiedene Bildverarbeitungsmethoden erklärt, welche typischerweise für die Bearbeitung medizinischer Bilder eingesetzt werden.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Noch erfolgreichem Besuch der Veranstaltung haben die Studierenden einen Überblick über die Funktionsweise und die Möglichkeiten der modernen medizinischen Bildverarbeitung. Studierende sind dazu in der Lage, einfache bis mittlere medizinische Bildverarbeitungsaufgaben selbständig zu lösen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Mathematische Grundlagen sind dringend empfehlenswert. Ferner wird empfohlen, die Vorlesung „Bildverarbeitung“ oder eine vergleichbare Veranstaltung vorher besucht zu haben.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0379-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				

	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0379-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> 1) Heinz Handels: Medizinische Bildverarbeitung 2) 2) Gonzalez/Woods: Digital Image Processing (last edition) 3) 3) Bernd Jähne: Digitale Bildverarbeitung. 6. überarbeitete und erweiterte Auflage. Springer, Berlin u. a. 2005, ISBN 3-540-24999-0. 4) Kristian Bredies, Dirk Lorenz: Mathematische Bildverarbeitung. Einführung in Grundlagen und moderne Theorie. Vieweg+Teubner, Wiesbaden 2011, ISBN 978-3-8348-1037-3.
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Ambient Intelligence					
<b>Modul Nr.</b> 20-00-0390	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0390-iv	Ambient Intelligence	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Die Vorlesung führt in aktuelle Entwicklungen von Ambient Intelligence ein. Im Vordergrund der Vorlesung steht die Mensch-Maschine-Interaktion (MMI) in intelligenten Umgebungen in einem allgegenwärtigen Informationsraum, wie sie beispielsweise zunehmend durch eingebettete Systeme in alltägliche Gebrauchsobjekte gegeben ist. Spezieller Fokus wird auf den mobilen Aspekt eines allgegenwärtigen Informationszugriffs und der Informationsaufbereitung und -darstellung in mobilen Endgeräten gelegt. Dabei soll einerseits ein Einblick in die grundlegenden Technologien, Anwendungen und Experimente gegeben werden und andererseits (nicht im Schwerpunkt) auch die sozio-kulturellen Implikationen und Aspekte neuer Ambient Intelligence Lösungen diskutiert werden. Zusätzliche Themen der Vorlesung sind System-Architekturen für verteilte Umgebungen, Kontext-Awareness und Kontext-Management, Benutzermodelle und deren Implikationen, Sensornetzwerke und Interaktionstechniken. Die Vorlesung wird Beispiele aktueller Projekte diskutieren und die internationalen Forschungslinien von Ambient Intelligence beleuchten.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung erfolgreich besucht haben, können sie Technologietrends und Forschungserkenntnisse im Bereich Ambient Intelligence beschreiben. Die wichtigsten Konzepte zur Realisierung „intelligenter Umgebungen“ - intelligente Netzwerke und Objekte, Techniken der erweiterten, mobilen Realität, ubiquitäre und allgegenwärtige Informationsräume, nomadische Kommunikationen, Echt-Zeit-Kommunikation und relevante Middleware, Eingebettete Systeme, Sensor Netzwerke und Wearable Computing - können diskutiert und eingeordnet werden. Nach Abschluss der zugehörigen Übung können Studierende die Projektphasen der Entwicklung einer Ambient-Intelligence Anwendung eigenständig planen und realisieren.				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch von “Visual Computing“ und „Multimodale Interaktion mit intelligenten Umgebungen“ oder vergleichbarer Veranstaltungen
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0390-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0390-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird jeweils passend zu den aktuellen Themen bekanntgegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Programmierung Massiv-Paralleler Prozessoren					
<b>Modul Nr.</b> 20-00-0419	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0419-iv	Programmierung Massiv-Paralleler Prozessoren	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> - Grundlagen massiv-paralleler Hardware mit einem Schwerpunkt auf modernen Beschleunigern - parallele Algorithmen - effiziente Programmierung massiv-paralleler Systeme - praktische Programmierprojekte mit Co-Betreuung durch einen Wissenschaftler aus seiner Anwendungsdomain				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung sind Studierende dazu in der Lage, Problemstellungen im Kontext massiv-paralleler Systeme zu analysieren. Sie können selbständig neue Anwendungen entwickeln und ihre Performanz systematisch verbessern. Sie verstehen grundlegende parallele Algorithmen und Programmierparadigmen und können sich selbständig aktuelle Literatur erarbeiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  solide Programmierkenntnisse in C/C++  Kenntnisse in paralleler Programmierung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0419-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0419-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Capturing Reality					
<b>Modul Nr.</b> 20-00-0489	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0489-iv	Capturing Reality	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Dieser Kurs deckt ein breites Spektrum von Techniken zur Digitalisierung und Modellierung unserer Welt mit einem Fokus auf Anwendungen in der Computergraphik und Computer Vision ab. Dies beinhaltet insbesondere: - grundlegende Werkzeuge und Kalibrationstechniken für die Digitalisierung - Digitalisierungs- und Modellierungstechniken für verschiedenste Objekt- und Szeneneigenschaften (z.B. Geometrie, Reflexionseigenschaften) - grundlegende mathematische Modellierungs- und Optimierungstechniken - Implementierung und praktische Anwendung einer Reihe von Techniken				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung sind Studierende dazu in der Lage, Digitalisierungs- und Modellierungsprobleme für Objekte und Szenen in Computergraphik und Computer Vision sowie die zugrunde liegenden Techniken zu analysieren. Sie können selbständig neue Versuchsaufbauten entwickeln, Experimente durchführen und die Ergebnisse auswerten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch der Veranstaltungen „Graphische Datenverarbeitung I“ oder „Computer Vision I“ oder vergleichbaren Veranstaltungen sowie grundlegende Programmierkenntnisse in C/C++				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0489-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				



	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0489-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Noriko Kurachi: The Magic of Computer Graphics. A K Peters/CRC Press Richard Szeliski: Algorithms and Applications, Springer Marcus Magnor, Oliver Grau, Olga Sorkine-Hornung, Christian Theobalt: Digital Representations of the Real World: How to Capture, Model, and Render Visual Reality Wolfgang Förstner, Bernhard P. Wrobel: Photogrammetric Computer Vision - Geometry, Orientation and Reconstruction
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> TK2: Human Computer Interaction					
<b>Modul Nr.</b> 20-00-0535	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0535-v1	TK2: Human Computer Interaction	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> Die Vorlesung stellt verschiedene grundlegende Konzepte, Modelle und Theorien aus dem Bereich der Human Computer Interaction (HCI) vor. Die Veranstaltung umfasst die folgenden Inhalte: <ul style="list-style-type: none"><li>• Theoretische Grundlagen aus Psychologie und Interaktionsgestaltung als Basis für die Gestaltung von Nutzerschnittstellen</li><li>• Überblick über verschiedene Typen von Nutzerschnittstellen</li><li>• Command-line interfaces</li><li>• Grafische Nutzerschnittstellen, u.a. Mac OS und Windows</li><li>• Interaktive Oberflächen, u.a. Tabletops, Multitouch</li><li>• Mobile user interfaces, u.a. basierend auf iPhone OS, Android</li><li>• Pen-based user interfaces, u.a. elektronische Stifte</li><li>• Tangible user interfaces, Organic user interfaces</li><li>• Sprachbasierte user interfaces</li><li>• Beurteilung, Messung, Bewertung von Nutzerschnittstellen</li><li>• Nutzerstudien</li><li>• Quantitative Evaluationsmethoden</li><li>• Qualitative Evaluationsmethoden</li><li>• Nutzerzentrierte Softwareentwicklung</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach der Teilnahme an dieser Lehrveranstaltung haben Studierende <ul style="list-style-type: none"><li>• Verständnis der psychologischen Grundlagen des Designs von Benutzerschnittstellen erworben</li><li>• Methoden des user-centric design process kennengelernt</li><li>• Überblickswissen über die gängigen UI Konzepte erworben</li></ul>				

	<ul style="list-style-type: none"> <li>• Evaluationstechniken kennen gelernt und angewandt</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b>
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0535-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0535-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein:  Ausgewählte Kapitel aus den folgenden Standardwerken: <ul style="list-style-type: none"> <li>• Donald Norman: The Design of Everyday Things</li> <li>• Alan Dix, Janet Finlay, Gregory Abowd and Russel Beale: Human-Computer Interaction</li> <li>• Jenny Preece , Yvonne Rogers and Helen Sharp: Interaction Design: Beyond Human-Computer Interaction</li> </ul>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Fortgeschrittener Compilerbau					
<b>Modul Nr.</b> 20-00-0701	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0701-v1	Fortgeschrittener Compilerbau	6	integrierte Veranstaltung	3
<b>2</b>	<b>Lerninhalt</b> - Compilierung und Laufzeitumgebung für objektorientierte Programmiersprachen - Kontrollflussgraphen als Zwischendarstellung - Statische Datenflußanalyse - Static Single Assignment Form - Eliminierung totaler und partieller Redundanz - Skalare Optimierung - Registerallokation - Ablaufplanung - Schleifenoptimierung - Aufbau realer Compiler (z.B. Phasen, Zwischendarstellung, Compilefluß)				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende verstehen nach erfolgreichem Besuch Techniken für die Übersetzung und Ausführung von objektorientierten Programmen auf Maschinenebene. Sie können die statische Datenflussanalyse auf Kontrollflussgraphen anwenden und sind geübt im praktischen Umgang mit deren SSA-Darstellung. Sie beherrschen Optimierungsverfahren für eine Reihe von Aufgaben sowie fundamentale Verfahren für die Registerallokation. Sie kennen die interne Struktur von realen Compilern für den Produktivbetrieb.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreicher Besuch der Veranstaltung “Einführung in den Compilerbau” oder vergleichbarer Veranstaltungen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0701-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0701-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein:</p> <p>Cooper/Torczon: Engineering a Compiler Muchnick: Advanced Compiler Design and Implementation Aho/Lam/Sethi/Ullman: Compilers - Principles, Techniques, and Tools</p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
User-Centered Design in Visual Computing					
<b>Modul Nr.</b> 20-00-0793	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0793-iv	User-Centered Design in Visual Computing	3	Integrierte Veranstaltung	2
<b>2</b>	<b>Lerninhalt</b> Die Entwicklung von benutzerzentrierten Softwarelösungen dient nicht nur zur besseren und effizienteren Nutzung von Software, sie erhöht vielmehr die Akzeptanz und somit auch die Verbreitung und Verwendung. Die Vorlesung “User Centered Design in Visual Computing“ richtet sich in erster Linie an Studierende des Fachbereichs Informatik und vermittelt Modelle, Methoden und Techniken zur benutzerzentrierten Entwicklung von Visualisierungssoftware und visuell-interaktiven Benutzerschnittstellen. Dabei werden insbesondere Methoden vorgestellt, die zu einer gesteigerten Akzeptanz und effizienterer Benutzung der entworfenen Lösungen führen. Des Weiteren werden Methoden der Evaluation vorgestellt, die die Akzeptanz und Nutzbarkeit messen. Die Vorlesung behandelt die eingeführten Themen mit besonderem Bezug zu Visual Computing und graphischen Benutzerschnittstellen. Stoffplan: <ul style="list-style-type: none"><li>• Usability</li><li>• User Experience</li><li>• Task Analysis</li><li>• Benutzerschnittstellen</li><li>• Interaktionsdesign</li><li>• Prototyping</li><li>• Graphikdesign und Informationsvisualisierung</li><li>• Evaluation während und nach der Softwareentwicklung</li><li>• Anwendungen</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende können nach erfolgreichem Besuch der Veranstaltung: <ul style="list-style-type: none"><li>• Geeignete Methoden zur Entwicklung von benutzerzentrierten Softwarelösungen identifizieren und begründen</li><li>• Techniken zu benutzerzentrierten Nutzungsschnittstellen anwenden</li><li>• Evaluationsmethoden zur Untersuchung der eingesetzten Techniken in den verschiedenen Phasen der Entwicklung identifizieren und auswählen</li><li>• Verbesserungen zur Informationsaufnahme und Navigation auf Basis vorhandener Untersuchungen und Evaluationen vorschlagen</li></ul>				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen des Visual Computing, wie sie beispielsweise in den Veranstaltungen „Visual Computing“ und „Graphische Datenverarbeitung I“ vermittelt werden
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0793-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0793-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Formale Spezifikation und Verifikation von Software					
<b>Modul Nr.</b> 20-00-0794	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0794-iv	Formale Spezifikation und Verifikation von Software	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b>  In dieser Vorlesung behandeln wir fortgeschrittene Themen aus dem Gebiet der formalen Spezifikation und deduktiven Verifikation objekt-orientierter Software.  Der Kurs deckt insbesondere folgende Themen ab: * Spezifikation von Interfaces und Klassen mit Hilfe von Queries, Ghost- und Modellfeldern; * Das "Framing" Problem: Statische und dynamische Frames * Programmlogik und -kalkül als Grundlage der deduktiven Verifikation * Spezifikation und Verifikation rekursiver Methoden und Schleifen * Modulare Verifikation: Sichtbarkeiten, Beweis und Anwendung von Framing-Eigenschaften * Automatische Erzeugung von Schleifeninvarianten und Methodenverträgen  Der Kurs behandelt vorwiegend sequentielle Programme. Es werden aber auch aktuelle Ansätze zur Spezifikation und Verifikation nebenläufiger bzw. verteilter Software diskutiert.  Für fast alle Themen wird deren praktische Anwendung mit Hilfe geeigneter Tools demonstriert und in den Übungen vertieft.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> * Erwerbung der Fähigkeit zur Spezifikation komplexer objekt-orientierter Software * Studierende sollen in der Lage sein einen für das vorliegende Problem passenden Spezifikationsansatz auszuwählen und anzuwenden * Studierende sollen in der Lage sein rekursive Methoden und Schleifen zu spezifizieren * Studierende sollen in der Lage sein mit Hilfe von deduktiver Verifikation ihre Programme als korrekt zu beweisen				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Grundlagenwissen über Logik erster Ordnung Inhalt der Vorlesungen „Aussagen- und Prädikatenlogik“ und „Formale Methoden im Softwareentwurf“ oder vergleichbarer Veranstaltungen				



<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0794-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0794-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Multithreading in C++					
Modul Nr. 20-00-0953	Leistungspunkte 10 CP	Arbeitsaufwand 300 h	Selbststudium 210 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0953-iv	Multithreading in C++	10	Integrierte Veranstaltung	6
2	Lerninhalt C++ bietet eine der fortschrittlichsten Threadschnittstellen, die heute verfügbar sind. Am Beispiel C++ führt dieser Kurs in die parallele Programmierung für gemeinsamen Speicher mit Threads ein. <ul style="list-style-type: none"><li>• Architekturen mit gemeinsamem Speicher</li><li>• Management von Threads</li><li>• Zugriff auf gemeinsame Daten</li><li>• Synchronisierung nebenläufiger Operationen</li><li>• Entwurf lockbasierter nebenläufiger Datenstrukturen</li><li>• Entwurf von nebenläufigem Code</li><li>• Testen und Fehlersuche</li></ul>				
3	Qualifikationsziele / Lernergebnisse Kompetenz in der Entwicklung paralleler Programme <ul style="list-style-type: none"><li>• Systematisch korrekte und effiziente parallele Programme entwickeln</li><li>• Parallele Datenstrukturen entwerfen und umsetzen</li></ul>				
4	Voraussetzung für die Teilnahme Empfohlen: Kenntnisse in C/C++				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0953-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0953-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Deep Learning für medizinische Bildgebung					
<b>Modul Nr.</b> 20-00-1014	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1014-iv	Deep Learning für medizinische Bildgebung	5	Integrierte Veranstaltung	3
<b>2</b>	<b>Lerninhalt</b> Formulierung der medizinischen Bildsegmentierung, Computergestützte Diagnostik und chirurgische Planung als Probleme des maschinellen Lernens, Deep Learning für medizinische Bildsegmentierung, Deep Learning für computergestützte Diagnostik, Chirurgische Planung von präoperativen Bildern mit Deep Learning, Tool-Präsenz Erkennung und Lokalisierung von endoskopischen Videos durch Deep Learning, Adversarial Beispiele für medizinische Bildgebung, Generative Adversarial Networks für Medizinische Bildgebung.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Abschluss des Kurses sind die Studierenden in der Lage sein, alle Komponenten der Formulierung eines medizinischen Bildanalyseproblems als Problem des Maschinellen Lernens zu verstehen. Sie sind auch in der Lage sein, fundierte Entscheidungen über die Wahl eines universellen Deep Learning Paradigmas für ein gegebenes medizinische Bildanalyseproblem zu treffen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  - Programmierkenntnisse - Verständnis des algorithmischen Designs - Kenntnisse aus dem Bereich Lineare Algebra - Der vorherige Besuche von „Bildverarbeitung“, „Computer Vision I“ und „Statistisches Maschinelles Lernen“ oder vergleichbarer Veranstaltungen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1014-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1014-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Tiefe Generative Modelle					
Modul Nr. 20-00-1035	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1035-iv	Tiefe Generative Modelle	6	Integrierte Veranstaltung	4
2	Lerninhalt Generative Modelle, implizite und explizite Modelle, Variational AutoEncoders, Generative Adversarial Networks, Numerische Optimierung für generative Modelle, Anwendungen in der medizinischen Bildverarbeitung				
3	Qualifikationsziele / Lernergebnisse Nachdem Studierende das Modul besucht haben, können sie - den Aufbau und die Funktionsweise Tiefer Generativer Modelle (Deep Generative Models, DGM) erklären - wissenschaftliche Veröffentlichungen zum Thema DGMs kritisch hinterfragen und damit fachlich beurteilen - grundlegende DGMs in einer dafür ausgelegten höheren Programmiersprache selbstständig konstruieren / implementieren - die Implementierung und Anwendung von DGMs auf unterschiedliche Anwendungen übertragen				
4	Voraussetzung für die Teilnahme Empfohlen:  - Programmierkenntnisse Python - Kenntnisse aus dem Bereich Lineare Algebra - Der vorherige Besuche von „Bildverarbeitung“, „Computer Vision I“ und „Statistisches Maschinelles Lernen“ oder vergleichbarer Veranstaltungen				
5	Prüfungsform Bausteinbegleitende Prüfung:  • [20-00-1035-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1035-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Wird in Veranstaltung bekannt gegeben.
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Verteilte Geometrieverarbeitung					
<b>Modul Nr.</b> 20-00-1075	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1075-iv	Verteilte Geometrieverarbeitung	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> * Grundlagen und Algorithmen der Geometrieverarbeitung: Smoothing, Remeshing, Delaunay-Triangulierung, Parametrisierung, Texturierung, u.a. * Einführung in Big Data und Cloud Computing * Indexstrukturen für den schnellen Zugriff auf massive Geometriedatenmengen: Quad tree, R-tree, Space-filling curves, u.a. * Verteilte und cloud-basierte Datenspeicherung * Architekturen für verteilte Verarbeitungspipelines * Programmiermodelle für verteilte Algorithmen (z.B. MapReduce) * Technologien und Frameworks für die verteilte Datenverarbeitung (z.B. Spark, Vert.x) und Geometrieverarbeitung (Draco, u.a.) * Deployment von verteilten Anwendungen in die Cloud * Ergänzend gibt es praktische und theoretische Übungen				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung besitzen die Studierenden grundlegende Kenntnisse zur Geometrieverarbeitung sowie zur verteilten, cloud-basierten Verarbeitung sehr großer Datenmengen im Allgemeinen. Sie sind in der Lage, selbstständig skalierbare Anwendungen zu entwickeln und diese in der Cloud auszuführen, um die Geometrieverarbeitung zu parallelisieren und damit die Performance zu erhöhen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  * Programmierkenntnisse in Java oder anderen JVM-Sprachen * Grundlegende Algorithmen und Datenstrukturen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1075-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)				



	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1075-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Model Checking					
Modul Nr. 20-00-1115	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1115-vl	Model Checking	3	Vorlesung	2
2	Lerninhalt * Temporallogiken: - Lineare temporal Logik (LTL), Computation Tree Logic (CTL) und CTL*: Syntax, Semantik, Komplexität * Modelprüfungsverfahren für LTL, CTL, CTL*, insbesondere Büchiautomaten * Partial Order Reduction * Timed Automata				
3	Qualifikationsziele / Lernergebnisse Im Rahmen des Kurses sollen die Studierenden folgende Fähigkeiten erwerben: * Verständnis der theoretischen Grundlagen der Temporallogiken LTL, CTL und CTL* * Fähigkeit zur Auswahl der geeigneten Logik zur Spezifikation und Modellprüfung in Abhängigkeit von dem zu modellierenden System und der zu prüfenden Eigenschaft * Verfahren und Techniken zur Modellprüfung (Model Checking) z.B. Modellprüfung mit Büchiautomaten, Partial Order Reduction u.ä. * Wissen über die Charakteristika und Grenzen der Modellprüfung * Kenntnisse in der Modellprüfung von Timed Automate * Fähigkeit zur Anwendung von Tools zur Modellprüfung				
4	Voraussetzung für die Teilnahme Empfohlen werden Kenntnisse in * Aussagenlogik * Deduktionssystemen * Automatentheorie				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1115-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%).</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1115-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Hands-On HCI					
Modul Nr. 20-00-1116	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1116-iv	Hands-On HCI	6	Integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Vielleicht haben Sie bereits von Virtual / Augmented Reality, 3D-Druck, am Körper getragenen oder anfassbaren (tangible) Benutzeroberflächen gehört oder diese sogar ausprobiert. Der Bereich Human-Computer-Interaktion (HCI) deckt all diese spannenden Themen ab und bietet die Möglichkeit, neue Prototypen zu bauen und diese in Benutzerstudien auszuprobieren. Wenn Sie Theorie und Praxis im Bereich der HCI verbinden möchten ist dieser Kurs - Hands-On HCI - genau das Richtige für Sie. Das Ziel des Kurses ist es, Sie durch den gesamten Forschungszyklus im Bereich der HCI zu führen. Damit kann dieser Kurs eine Vorbereitung für Ihre zukünftige Bachelor- / Masterarbeit in diesem Bereich sein, sowie einen ersten Baustein auf Ihrem akademischen Weg darstellen.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach Abschluss des Moduls können Studierende  - drei Ansätze zur HCI-Forschung voneinander unterscheiden und anwenden. - drei Arten empirischer Untersuchungen unterscheiden. - effektiv eine wissenschaftliche Publikation lesen. - zwischen Arten von HCI-Beiträgen unterscheiden. - Forschungsfragen, Hypothesen und experimentelle Variablen formulieren und definieren. - basierend auf den zuvor erarbeiteten Forschungsfragen ein dazu passendes Studiendesign entwerfen. - eine Studie durchführen und dabei quantitative und qualitative Methoden zur Datensammlung verwenden. - quantitative Daten auf der Basis von statistischen Methoden analysieren, auswerten und interpretieren. - qualitative Daten auf der Basis von Grounded Theory analysieren und interpretieren. - den Peer-Review Prozess verstehen und sowie Reviews für eine wissenschaftliche Publikation schreiben.				

	<ul style="list-style-type: none"> <li>- Evaluationstechniken mit und ohne Nutzern verstehen und anwenden.</li> <li>- die gewonnenen Erkenntnisse als wissenschaftliche Publikation verschriftlichen und vor einem Fachpublikum präsentieren.</li> </ul>
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen wird die vorherige Belegung von „TK2: Human-Computer Interaction“ oder einer vergleichbaren Veranstaltung.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1116-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1116-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Mensch- und Identitätsfokussiertes Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-1118	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1118-iv	Mensch- und Identitätsfokussiertes Maschinelles Lernen	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b>				
	Hintergründe und Konzepte von Human-Centric Machine Learning: Das Ziel von Identität und Human-Centric Machine Learning. Die Unterschiede zwischen Identitätslernen und anderen gängigen Klassifikationsarten.				
	Repräsentationsextraktion für subjektbezogene Daten: Methoden für die Feature Erstellung für identitätsbezogene Anwendungen. Grundlagen und Hintergründe für handgefertigte oder Deep Learning Features.				
	Deep-Learning Strategien für Identitätsrepräsentationen: Erlernen von Identitätsrepräsentationen mit Hilfe von Deep Learning. Lernstrategien und Loss-Funktionen.				
	Netzwerkarchitekturen und identitätsspezifische Komponenten.				
	Knowledge Transfer und Distillation: Transfer Learning und Identitätsrepräsentation. Konzepte und Anwendungen von Knowledge Distillation.				
	Effizientes Machine Learning: Beziehung zwischen Ressourcenbeschränkungen, Green-AI und Deep Learning. Methoden zum Aufbau effizienter Lösungen für Maschinelles Lernen.				
	Synthetische Identität: Die Notwendigkeit einer synthetischen Identität. Synthetische Identität als Adversarial. Generierung synthetischer identitätsgesteuerter Daten unter verschiedenen Einschränkungen.				
	Machine Learning Biases: Analyse der demografischen Fairness und der Ursachen der Fairnessprobleme. ML-basierte Abmilderung von demografischen Bias.				
	Privatsphäre erlernen: Analyse von unbeabsichtigt gelernten Informationen. Lernstrategien zur gezielten Unterdrückung von Informationen auf verschiedenen Repräsentationsebenen.				

	<p>Data Utility: Verständnis der Auswirkungen von Data Utility im Lernprozess. Verstehen von Sample Utility im Betrieb. ML-Konzepte und Strategien zur Schätzung von Sample Utilities.</p> <p>Angriffe auf Sample-Level: Überblick über Adversarial, Sample Manipulation und andere Angriffe auf Human-Centric ML. Deep Learning Konzepte, Netzwerklöcke und LossStrategien um Sample-Level Angriffe zu erkennen und zu umgehen.</p> <p>Explainability: Überblick über den Bedarf von Explainability in verschiedenen Entscheidungsprozessen. Verschiedene Strategien um Explainability für Themen aus vergangenen Vorlesungen.</p>
<b>3</b>	<p><b>Qualifikationsziele / Lernergebnisse</b></p> <p>Nach erfolgreichem Besuch des Kurses sind die Studierenden mit Konzepten des maschinellen Lernens im Umgang mit personen- und identitätsbezogenen Informationen vertraut. Sie verstehen die grundlegenden Techniken für die Extraktion subjektspezifischer Repräsentationen, einschließlich der damit verbundenen Konzepte für Knowledge Transfer und Distillation. Die Studierenden haben ein Verständnis für demografisch bedingte Verzerrungen beim maschinellen Lernen und Datenschutzbedenken zu Function-Creep erlangt, einschließlich der wichtigsten Konzepte zur Abschwächung dieser Probleme. Sie kennen die Anforderungen und Techniken, die für ein eingebettetes und effizientes HumanCentric Machine Learning erforderlich sind. Ebenfalls sind sie mit den Auswirkungen von Data Utility im Lernprozess und dem Hauptkonzept zur Schätzung der Utility von subjektbezogenen Daten vertraut. Sie werden fundiertes Wissen über die Erklärungsmethoden für ML-Entscheidungen auf der Grundlage von identitätsbezogenen Daten erlangen. Die Studierenden werden in die Konzepte der KI-Ethik und der KIRegulierung im Zusammenhang mit der Verarbeitung und Speicherung personenbezogener Daten eingeführt. Sie sind in der Lage, diese Techniken zur Lösung grundlegender Aufgaben im Bereich von Identitäts- und Human-Centric Machine Learning auf realistische Probleme anzuwenden.</p>
<b>4</b>	<p><b>Voraussetzung für die Teilnahme</b></p> <p>Empfohlen: Der vorherige Besuch der Veranstaltung „Visual Computing“ oder einer vergleichbaren Veranstaltung. Grundlagen in Mathematik und Wahrscheinlichkeitsrechnung.</p>
<b>5</b>	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1118-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1118-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

Modulname					
Programmanalyse					
Modul Nr. 20-00-1122	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Wahlbereich Software und Hardware		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1122-v1	Programmanalyse	6	Vorlesung und Übung	4
2	<b>Lerninhalt</b>  Statische Analysen sind Werkzeuge die Informationen von Programmen extrahieren ohne diese auszuführen. Statische Analysen haben vielseitige Anwendungen in integrierten Entwicklungsumgebungen (IDEs), Compilern und Continous Integration Servern. Zum Beispiel werden statische Analysen in IDEs verwendet um Programmfehler und Sicherheitslücken zu erkennen. Des Weiteren werden sie in Compilern zum Typechecking und für Optimierungen verwendet.  Dieser Kurs gibt einen Überblick über die zugrundeliegende Konzepte von statischen Analysen. Insbesondere diskutieren wir den Kompromiss zwischen der Performanz, der Präzision, und der Korrektheit von statischen Analysen. Des Weiteren werden Ihnen verschieden Sorten von statischen Analysen vorgestellt, wie zum Beispiel Kontrollflussanalysen, Datenflussanalysen, Zeigeranalysen, sowie Seiteneffekt- und Unveränderlichkeitsanalysen. Abschließend lernen sie verschiedene Analyseframeworks kennen, wie das monotone Framework, Big-Step Abstrakte Interpreter und IFDS/IDE Frameworks.  In der begleitenden Übung wenden Sie die neuen Analysekonzepte praktisch an, und erweitern oder entwickeln existierende Analysen.				
3	<b>Qualifikationsziele / Lernergebnisse</b>  Das Ziel dieses Kurses ist es die zugrundeliegenden Konzepte von statischen Analysen zu vermitteln. Dies erlaubt es Studierenden statische Analysen in IDEs effektiver zu verwenden. Des Weiteren sind Studierende nach dem Kurs in der Lage Eigenschaften von statischen Analysen wie Performanz und Präzision zu beurteilen				
4	<b>Voraussetzung für die Teilnahme</b>  Empfohlen: Wir erwarten, dass Studierende die Konzepte der Programmiersprachen, wie Zuweisungen, Schleifen, Ausnahmebehandlung, Objekte, und anonyme Funktionen gut verstanden haben. Des Weiteren sollten die Kursteilnehmer*innen mit grundlegender Universitätsmathematik und Logik vertraut sein.				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1122-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1122-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Wahlbereich Theorie**  
**(Theoretische Informatik)**

## Modulbeschreibung

<b>Modulname</b>					
Statische und dynamische Programmanalyse					
<b>Modul Nr.</b> 20-00-0580	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Theorie		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0580-iv	Statische und dynamische Programmanalyse	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> - operationelle Semantiken für sequentielle und parallele Programme - Übersicht über Techniken zur statischen und dynamischen Programmanalyse - Abstrakte Interpretation - Datenflussanalysen - Slicing-Techniken - typbasierte Programmanalysen - Konzepte der Laufzeitüberwachung - Implementierungstechniken zur Laufzeitüberwachung - Sprachbasierte Sicherheit - Korrektheit und Präzision von Programmanalysen				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung kennen Studierende ein Spektrum von unterschiedlichen Programmanalysen. Sie verstehen die Funktionsweise der einzelnen Analysetechniken und verstehen die Unterschiede zwischen diesen. Sie können beurteilen, welche Analysetechnik für welche Problemstellung in Frage kommt und haben die Fähigkeit, die ausgewählte Analysetechnik einzusetzen. Sie können Programmanalysen bezüglich ihrer Präzision und Korrektheit beurteilen. Sie können Programmanalysen auch implementieren und Varianten von bekannten Programmanalysen definieren.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Informatik- und Mathematikkenntnisse entsprechend den ersten 4 Semestern des Bachelorstudiengangs Informatik, insbesondere grundlegende Logikkenntnisse und Fähigkeit, mit formalen Sprachen und Kalkülen umzugehen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-0580-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0580-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Automatische Softwareverifikation					
<b>Modul Nr.</b> 20-00-1069	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Theorie		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1069-iv	Automatische Softwareverifikation	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Die Veranstaltung befasst sich mit dem Techniken zur automatischen Softwareverifikation und behandelt dabei folgende Themebereiche: - operationelle Semantik von sequentiellen Programmen - konfigurierbare Programmanalyse inklusive Konfiguration für Datenflussanalysen und Model Checking - counter-example guided abstraction refinement (CEGAR) - Bounded Model Checking - k-Induktion - kooperative Verifikation, insbesondere Conditional Model Checking - inkrementelle Verifikation - Nachprüfung von Verifikationsergebnissen (a la Proof-Carrying Code, Witness Validation) - Generierung von Testeingaben mittels Verifizierern				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden eine Vielzahl von Verfahren zur automatischen Verifikation benennen. Sie können die den Verfahren zugrunde liegenden Formalismen wiedergeben, die Funktionsweise der Verfahren beschreiben und die Verfahren klassifizieren. Außerdem können die Studierenden die Verfahren auf Beispielen anwenden und neue konfigurierbare Programmanalysen entwickeln.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Informatikkenntnisse entsprechend der ersten vier Semester des Bachelorstudiengangs Informatik, insbesondere Kenntnisse aus der Vorlesung „Aussagen- und Prädikatenlogik“ oder Vergleichbares.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1069-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1069-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Verifikation paralleler Programme					
<b>Modul Nr.</b> 20-00-1079	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Theorie		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1079-iv	Verifikation paralleler Programme	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Die Veranstaltung befasst sich mit überwiegend automatischen Techniken zur Verifikation von parallelen Programmen, insbesondere multi-threaded Programmen mit gemeinsamen Speicher. Die Veranstaltung behandelt dabei folgende Themenbereiche: - Semantik von parallelen Programmen (z.B. Interleaving-Semantik, Semantik von ausgewählten schwachen Speichermodellen) - Statische und dynamische Techniken zur Erkennung von Data Races - Techniken der Deadlockanalyse - Analyse von Programmeigenschaften (z.B. mittels Sequentialisierung, Bounded Model Checking, etc.) - Partial Order Reduction - Thread-modulare Verifikation - Verifikation unter schwachen Speichermodellen				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden Verfahren zur Verifikation von parallelen Programmen, insbesondere Verfahren zur Analyse von Data Races, Deadlocks und Sicherheitseigenschaften (Safety) benennen. Sie können die den Verfahren zugrunde liegenden Formalismen wiedergeben, die Funktionsweise der Verfahren beschreiben und die Verfahren auf Beispielen anwenden. Außerdem können die Studierenden die Stärken und Schwächen der Verfahren beurteilen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Informatikkenntnisse entsprechend der ersten vier Semester des Bachelorstudiengangs Informatik. Vorteilhaft, aber nicht erforderlich ist der Besuch der Veranstaltung „Automatische Software Verifikation“.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1079-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				



	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1079-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Einführung in das Quantencomputing					
<b>Modul Nr.</b> 20-00-1136	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Wahlbereich Theorie		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1136-iv	Einführung in das Quantencomputing	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Allgemeine Einführung und Motivation Einführung in die Quantenmechanik (Zustände, Messungen, Evolution, ein kurzer Überblick zur linearen Algebra) Elementare Quantengatter und Schaltkreismodell Universelle Quantenberechnungen Quantenparallelismus und der Deutsch-Jozsa-Algorithmus Simon's Algorithm Die Fourier-Transformation Der Shor-Algorithmus Das Problem der versteckten Untergruppe Der Grover-Algorithmus Quantenfehlerkorrektur und Fehlertoleranz Verschränkung und Nichtlokalität Eine grundlegende Einführung in die Quantenschlüsselverteilung Überblick über Quantencomputerplattformen und Aussagen zur Quantenüberlegenheit				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Abschluss des Kurses sind die Studierenden mit allen grundlegenden Konzepten der Quanteninformationsverarbeitung und -berechnung vertraut und können diese mit der Quantenprogrammiersprache Qiskit programmieren. Sie lernen die wichtigsten "Eigenheiten" der Quantenwelt kennen und können diese mit rechnerischen und kryptographischen Aufgabenstellungen verbinden. Am Ende der Vorlesung wird eine Zusammenfassung der neuesten Entwicklungen in Industrie und Wissenschaft gegeben, die es den Studierenden ermöglicht, ihre zukünftigen Interessen in diesem Bereich zu steuern.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen werden grundlegende Kenntnisse in elementarer linearer Algebra (Matrixmultiplikation, Ermittlung von Eigenwerten)				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1136-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Empfohlen werden grundlegende Kenntnisse in elementarer linearer Algebra (Matrixmultiplikation, Ermittlung von Eigenwerten)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1136-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Data Science and Engineering**

**Wahlbereich Data Science Applications**

## Modulbeschreibung

<b>Modulname</b> Computer Vision					
<b>Modul Nr.</b> 20-00-0157	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0157-iv	Computer Vision	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Grundlagen der Bildformierung</li><li>• Lineare und (einfache) nichtlineare Bildfilterung</li><li>• Grundlagen der Mehransichten-Geometrie</li><li>• Kamerakalibrierung &amp; -posenschätzung</li><li>• Grundlagen der 3D-Rekonstruktion</li><li>• Grundlagen der Bewegungsschätzung aus Videos</li><li>• Template- und Unterraum-Ansätze zur Objekterkennung</li><li>• Objektklassifikation mit Bag of Words</li><li>• Objektdetektion</li><li>• Grundlagen der Bildsegmentierung</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Studierende beherrschen nach erfolgreichem Besuch der Veranstaltung die Grundlagen der Computer Vision. Sie verstehen grundlegende Techniken der Bild- und Videoanalyse, und können deren Annahmen und mathematische Formulierungen benennen, sowie die sich ergebenden Algorithmen beschreiben. Sie sind in der Lage diese Techniken praktisch so umzusetzen, dass sie grundlegende Bildanalyseaufgaben an Hand realistischer Bilddaten lösen können.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Der vorherige Besuch von „Visual Computing“ oder einer vergleichbaren Veranstaltung</p>				
5	<b>Prüfungsform</b> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"><li>• [20-00-0157-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0157-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning  M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden regelmässig aktualisiert und beinhalten beispielsweise:</p> <ul style="list-style-type: none"> <li>• R. Szeliski, "Computer Vision: Algorithms and Applications", Springer 2011</li> <li>• D. Forsyth, J. Ponce, "Computer Vision -- A Modern Approach", Prentice Hall, 2002</li> </ul>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Computer Vision II					
<b>Modul Nr.</b> 20-00-0401	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0401-iv	Computer Vision II	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Computer Vision als (probabilistische) Inferenz</li><li>• Robuste Schätzung und Modellierung</li><li>• Grundlagen der Bayes'schen Netze und Markov'schen Zufallsfelder</li><li>• Grundlegende Inferenz- und Lernverfahren der Computer Vision</li><li>• Bildrestaurierung</li><li>• Stereo</li><li>• Optischer Fluß</li><li>• Bayes'sches Tracking von (artikulierten) Objekten</li><li>• Semantische Segmentierung</li><li>• Aktuelle Themen der Forschung</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Studierende haben nach erfolgreichem Besuch der Veranstaltung ein vertieftes Verständnis der Computer Vision. Sie formulieren Fragestellungen der Bild- und Videoanalyse als Inferenzprobleme und berücksichtigen dabei Herausforderungen reeller Anwendungen, z.B. im Sinne der Robustheit. Sie lösen das Inferenzproblem mittels diskreter oder kontinuierlicher Inferenzalgorithmen, und wenden diese auf realistische Bilddaten an. Sie evaluieren die anwendungsspezifischen Ergebnisse quantitativ.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Der vorherige Besuch von „Visual Computing“ und „Computer Vision I“ oder vergleichbaren Veranstaltungen ist empfohlen.</p>				
5	<b>Prüfungsform</b> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"><li>• [20-00-0401-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0401-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning  M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden regelmässig aktualisiert und beinhalten beispielsweise:</p> <ul style="list-style-type: none"> <li>• S. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012</li> <li>• R. Szeliski, "Computer Vision: Algorithms and Applications", Springer 2011</li> </ul>
10	<p><b>Kommentar</b></p>



## Modulbeschreibung

<b>Modulname</b> Natural Language Processing and the Web					
<b>Modul Nr.</b> 20-00-0433	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0433-iv	Natural Language Processing and the Web	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Das Web beinhaltet mehr als 10 Milliarden indexierbare Webseiten, die mittels Stichwortsuche zugänglich sind. Die Vorlesung behandelt Methoden der automatischen Sprachverarbeitung bzw. des Natural Language Processing (NLP) zur Verarbeitung großer Mengen unstrukturierter Texte im Web und zur Analyse von Online-Inhalten als wertvolle Ressource für andere sprachtechnologische Anwendungen im Web.  Zentrale Inhalte: <ul style="list-style-type: none"><li>• Verarbeitung unstrukturierter Texte im Web<ul style="list-style-type: none"><li>○ NLP-Grundlagen: Tokenisierung, Wortartenerkennung, Stemming, Lemmatisierung, Chunking</li><li>○ UIMA: Grundlagen und Anwendungen</li><li>○ Web-Inhalte und ihre Charakteristika, u.a. verschiedene Genres, z.B. persönliche Seiten, Nachrichtenportale, Blogs, Foren, Wikis</li><li>○ Das Web als Korpus, insb. innovative Verwendung des Webs als sehr großes, verteiltes, verlinktes, wachsendes und multilinguales Korpus</li></ul></li><li>• NLP-Anwendungen für das Web<ul style="list-style-type: none"><li>○ Einführung in das Information Retrieval</li><li>○ Web-Suche und natürlichsprachliche Suchschnittstellen</li><li>○ Web-basierte Beantwortung von natürlichsprachlichen Fragen</li><li>○ Web-Mining im Web 2.0, z.B. Wikipedia, Wiktionary</li><li>○ Qualitätsbewertung von Web-Inhalten</li><li>○ Multilingualität</li><li>○ Internet-of-Services: Service Retrieval</li><li>○ Sentimentanalyse und Community Mining</li><li>○ Paraphrasen, Synonyme, semantische Verwandtschaft und das Web</li></ul></li></ul>				

3	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung besucht haben, können sie <ul style="list-style-type: none"> <li>• Methoden und Ansätze zur Verarbeitung unstrukturierter Texte verstehen und differenzieren,</li> <li>• die Arbeitsweise von Web-Suchmaschinen nachvollziehen und erläutern,</li> <li>• exemplarische Anwendungen der Sprachverarbeitung im Web selbständig aufbauen und analysieren,</li> <li>• das Potenzial von Web-Inhalten für die Verbesserung von sprachtechnologischen Anwendungen analysieren und einschätzen.</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlegende Kenntnisse über Algorithmen und Datenstrukturen sowie Programmierkenntnisse in Java werden erwartet
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0433-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0433-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.

## Modulbeschreibung

<b>Modulname</b> Lernende Roboter					
<b>Modul Nr.</b> 20-00-0629	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0629-v1	Lernende Roboter	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> - Grundlagen aus der Robotik und des Maschinellen Lernens für Lernende Roboter - Maschinellen Lernen von Modellen - Representation einer Policy. Hierarchische Abstraktion mit Bewegungsprimitiven - Imitationslernen - Optimale Steuerung mit gelernten Modellen - Reinforcement Learning und Policy Search-Verfahren - Inverses Reinforcement Learning				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichen Abschluss der Lehrveranstaltung verstehen Studierende die Grundlagen des Maschinellen Lernens und der Robotik. Sie können maschinelle Lernverfahren anwenden um einen Roboter zu befähigen, neue Aufgaben zu erlernen. Studierende verstehen die Grundlagen von Reinforcement Learning und können verschiedene Algorithmen anwenden um eine Policy des Roboters aufgrund von Interaktion mit der Umgebung zu erlernen. Sie verstehen den Unterschied zwischen Imitation Learning, Reinforcement Learning, Policy Search und Inverse Reinforcement Learning und können einschätzen, wann sie welchen Ansatz verwenden sollen. Sie können diese Ansätze auch problemlos auf geeignete Aufgabenstellungen anwenden.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Gute Programmierkenntnisse in Matlab und der vorherige Besuch von „Statistisches Maschinelles Lernen“ oder einer vergleichbaren Veranstaltung sind hilfreich aber nicht zwingend erforderlich				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0629-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0629-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Deisenroth, M. P.; Neumann, G.; Peters, J. (2013). A Survey on Policy Search for Robotics, Foundations and Trends in Robotics Kober, J; Bagnell, D.; Peters, J. (2013). Reinforcement Learning in Robotics: A Survey, International Journal of Robotics Research C.M. Bishop, Pattern Recognition and Machine Learning (2006), R. Sutton, A. Barto. Reinforcement Learning - an Introduction Nguyen-Tuong, D.; Peters, J. (2011). Model Learning in Robotics: a Survey
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Deep Learning für medizinische Bildgebung					
Modul Nr. 20-00-1014	Leistungspunkte 5 CP	Arbeitsaufwand 150 h	Selbststudium 105 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1014-iv	Deep Learning für medizinische Bildgebung	5	Integrierte Veranstaltung	3
2	Lerninhalt Formulierung der medizinischen Bildsegmentierung, Computergestützte Diagnostik und chirurgische Planung als Probleme des maschinellen Lernens, Deep Learning für medizinische Bildsegmentierung, Deep Learning für computergestützte Diagnostik, Chirurgische Planung von präoperativen Bildern mit Deep Learning, Tool-Präsenz Erkennung und Lokalisierung von endoskopischen Videos durch Deep Learning, Adversarial Beispiele für medizinische Bildgebung, Generative Adversarial Networks für Medizinische Bildgebung.				
3	Qualifikationsziele / Lernergebnisse Nach erfolgreichem Abschluss des Kurses sind die Studierenden in der Lage sein, alle Komponenten der Formulierung eines medizinischen Bildanalyseproblems als Problem des Maschinellen Lernens zu verstehen. Sie sind auch in der Lage sein, fundierte Entscheidungen über die Wahl eines universellen Deep Learning Paradigmas für ein gegebenes medizinische Bildanalyseproblem zu treffen.				
4	Voraussetzung für die Teilnahme Empfohlen:  - Programmierkenntnisse - Verständnis des algorithmischen Designs - Kenntnisse aus dem Bereich Lineare Algebra - Der vorherige Besuche von „Bildverarbeitung“, „Computer Vision I“ und „Statistisches Maschinelles Lernen“ oder vergleichbarer Veranstaltungen				
5	Prüfungsform Bausteinbegleitende Prüfung:  • [20-00-1014-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1014-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Deep Learning: Architectures & Methods					
Modul Nr. 20-00-1034	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1034-iv	Deep Learning: Architectures & Methods	6	Integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Auffrischung des Hintergrundwissens</li><li>• Deep Feedforward Netze</li><li>• Regularisierung im Deep Learning</li><li>• Optimierung zum Training tiefer Netze</li><li>• Convolutional tiefe Netze</li><li>• Modellierung von Sequenzen durch Rekordernte und Rekursive Netze</li><li>• Lineare Faktor Modelle</li><li>• Autoenkoder</li><li>• Repräsentationslernen</li><li>• Strukturierte Probabilistische Modelle zum Deep Learning</li><li>• Monte Carlo Methoden</li><li>• Approximative Inferenz</li><li>• Tiefe generative Modelle</li><li>• Deep Reinforcement Learning</li><li>• Deep Learning in Vision</li><li>• Deep Learning in NLP</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Dieser Kurs richtet sich an Studierende mit fortgeschrittenem Erfahrung im maschinellen Lernen und vermittelt diesen Studierenden das notwendige Wissen, um eigenständig Forschungsprojekte im Bereich der Deep Learning durchzuführen, z.B. im Rahmen einer Bachelor- oder Masterarbeit. Dies betrifft sowohl ein grundlegendes Verständnis der algorithmischen Ansätze zum Deep Learning als auch die der Architekturen der tiefen tiefen Netze.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Der vorherige Besuch von „Statistisches Maschinelles Lernen“ und „Data Mining und Maschinelles Lernen“ oder vergleichbarer Veranstaltungen</p>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1034-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1034-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b>					
Reinforcement Learning: Von Grundlagen zu den tiefen Ansätzen					
<b>Modul Nr.</b> 20-00-1047	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1047-iv	Reinforcement Learning: Von Grundlagen zu den tiefen Ansätzen	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Auffrischung des Hintergrundwissens</li><li>• Black box Reinforcement Learning</li><li>• Modellierung als Bandit, Markov Decision Processes und Partially Observable Markov Decision Processes</li><li>• Optimale Steuerung und Regelung</li><li>• Modellernen</li><li>• Wertefunktionslernen</li><li>• Policy Search</li><li>• Tiefe Wertefunktion Methoden</li><li>• Tiefe Policy Search Methoden</li><li>• Exploration vs Exploitation</li><li>• Hierarchisches Reinforcement Learning</li><li>• Intrinsische Motivation</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Dieser Kurs richtet sich an Studierende mit erster Erfahrung im maschinellen Lernen und vermittelt diesen Studierenden das notwendige Wissen, um eigenständig Forschungsprojekte im Bereich der Reinforcement Learning durchzuführen, z.B. im Rahmen einer Bachelor- oder Masterarbeit. Dies betrifft sowohl ein grundlegendes Verständnis der algorithmischen Ansätze zum Reinforcement Learning als auch Anwendungen von tiefen Netzen.</p>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen:</p> <p>Gute Programmierkenntnisse in Python. Der vorherige Besuch von „Statistisches Maschinelles Lernen“ oder einer vergleichbaren Veranstaltung ist hilfreich aber nicht zwingend erforderlich</p>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1047-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1047-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Data Science and Engineering**

**Wahlbereich Data Systems Engineering**

## Modulbeschreibung

<b>Modulname</b> Informationsvisualisierung und Visual Analytics					
<b>Modul Nr.</b> 20-00-0294	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0294-iv	Informationsvisualisierung und Visual Analytics	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Diese Vorlesung wird eine detaillierte Einführung in die Informationsvisualisierung geben, um sich dann intensiv den wissenschaftlichen Fragestellungen und praxisnahen Anwendungsszenarien von Visual Analytics zu widmen. •Überblick der Informationsvisualisierung und Visual Analytics (Definitionen, Modelle, Historie) •Datenpräsentierung und Datentransformation •Abbildung von Daten auf visuelle Strukturen • Visuelle Repräsentierungen und Interaktion fuer bivariate, multivariate Daten, Zeitreihen, Graphen und Geographische Daten • Grundlagen von Data Mining •Grundlagen von Visual Analytics: - Analytische Beweisführung - Data Mining • Evaluation von Visual Analytics Systemen  Anwendungsgebiete: Medizin, Biologie, Finanzen und Wirtschaft, Meteorologie, Rettungsdienst,....				
3	<b>Qualifikationsziele / Lernergebnisse</b> Studierende können nach erfolgreichem Besuch der Veranstaltung  •Informationsvisualisierungsmethoden für verschiedene Datentypen benutzen  •interactive Visualisierungssysteme für Daten aus verschiedenen Anwendungsgebieten designen  •Visualisierung und automatische Datenverarbeitung kombinieren um Big Data Probleme zu lösen				

	<ul style="list-style-type: none"> <li>• Wissen über Hauptcharakteristika menschlicher visuellen Wahrnehmung in Informationsvisualisierung und Visual Analytics anwenden</li> <li>• geeignete Evaluationsmethode für spezifische Situationen und Szenarien auswählen</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Interesse an Methoden der Computergrafik und Visualisierung
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0294-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0294-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird in der Vorlesung bekanntgegeben. Beispiele für verwendete Literatur könnten sein: C. Ware: Information Visualization: Perception for Design Ellis et al: Mastering the Information Age
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Software Engineering - Design and Construction					
<b>Modul Nr.</b> 20-00-0341	<b>Leistungspunkte</b> 8 CP	<b>Arbeitsaufwand</b> 240 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0341-iv	Software Engineering - Design and Construction	8	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Der primäre Inhalt der Veranstaltung ist der Entwurf modularer Software, um wartbare, wiederverwendbare und erweiterbare Softwaresysteme zu erhalten. Integraler Bestandteil der Veranstaltung ist die Diskussion der Beziehung zwischen den Eigenschaften fortschrittlicher Programmiersprachen und dadurch möglicher Entwurfsalternativen. Weiterhin wird die Auswirkung der Programmiersprache auf den Entwurf eines Softwaresystems als Ganzes besprochen. Die Vorlesung behandelt insbesondere: <ul style="list-style-type: none"><li>• Prinzipien des Klassenentwurfs unter Verwendung fortgeschrittener Entwurfsmuster und fortschrittlicher Programmiersprachen;</li><li>• Prinzipien des Entwurfs auf Paketebene;</li><li>• Architekturelle Stile;</li><li>• Dokumentation des Entwurfs;</li><li>• Refactorings existierender Software;</li><li>• Metriken zur Evaluierung von Entwürfen.</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Abschluss der Lehrveranstaltung sind Studierende in der Lage die folgenden Aufgaben durchzuführen: <ul style="list-style-type: none"><li>• Sie können den Entwurf existierender Systeme in Hinblick auf ihre Modularität analysieren und ggf. Refactorings vorschlagen, die der Verbesserung bzw. Wiederherstellung selbiger dienen.</li><li>• Sie verstehen die mittel- und langfristigen Auswirkung nicht-modularer Softwaresysteme.</li><li>• Sie kennen fortgeschrittene Entwurfsmuster und können diese in existierendem Code identifizieren und auch einsetzen, um neue Probleme zu lösen.</li><li>• Sie kennen etablierte architekturelle Stile und können diese einsetzen.</li></ul>				

	<ul style="list-style-type: none"> <li>• Sie verstehen, dass die Lösung eines Entwurfsproblems von der gewählten Programmiersprache abhängt und sind in der Lage entsprechende Entscheidungen kritisch zu hinterfragen.</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der erfolgreiche Besuch der Veranstaltung „Software Engineering“ oder einer vergleichbaren Veranstaltung
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0341-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0341-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> <ul style="list-style-type: none"> <li>• Bass, L.; Clements, P.; Kazman, R. ; Software Architecture in Practice, Addison-Wesley</li> <li>• Booch, G. Object-Oriented Analysis and Design with Applications. Addison-Wesley</li> <li>• Budd, T. Introduction to Object-Oriented Programming. 2nd. ed., Addison-Wesley</li> <li>• Buschmann, F. et al. Pattern-Oriented Software Architecture: A System of Patterns. John Wiley &amp; Sons.</li> <li>• Czarnecki, K. and Eisenecker, U. Generative Programming. Addison-Wesley.</li> <li>• Garland, D. and Shaw, M. Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall.</li> <li>• Gamma, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.</li> <li>• Martin, Robert. Agile Software Development. Principles, Patterns, and Practices. Pearson US Imports &amp; PHIPEs.</li> <li>• Riel, A. Object-Oriented Design Heuristics. Addison-Wesley.</li> </ul>

<b>10</b>	<b>Kommentar</b>
-----------	------------------



## Modulbeschreibung

<b>Modulname</b>					
Multithreading in C++					
<b>Modul Nr.</b> 20-00-0953	<b>Leistungspunkte</b> 10 CP	<b>Arbeitsaufwand</b> 300 h	<b>Selbststudium</b> 210 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0953-iv	Multithreading in C++	10	Integrierte Veranstaltung	6
<b>2</b>	<b>Lerninhalt</b> C++ bietet eine der fortschrittlichsten Threadschnittstellen, die heute verfügbar sind. Am Beispiel C++ führt dieser Kurs in die parallele Programmierung für gemeinsamen Speicher mit Threads ein. <ul style="list-style-type: none"><li>• Architekturen mit gemeinsamem Speicher</li><li>• Management von Threads</li><li>• Zugriff auf gemeinsame Daten</li><li>• Synchronisierung nebenläufiger Operationen</li><li>• Entwurf lockbasierter nebenläufiger Datenstrukturen</li><li>• Entwurf von nebenläufigem Code</li><li>• Testen und Fehlersuche</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Kompetenz in der Entwicklung paralleler Programme <ul style="list-style-type: none"><li>• Systematisch korrekte und effiziente parallele Programme entwickeln</li><li>• Parallele Datenstrukturen entwerfen und umsetzen</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Kenntnisse in C/C++				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0953-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				

6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0953-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Fortgeschrittenes Multithreading in C++					
<b>Modul Nr.</b> 20-00-0977	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0977-iv	Fortgeschrittenes Multithreading in C++	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> C++ bietet eine der modernsten Threadschnittstellen, die heute verfügbar sind. Am Beispiel C++ führt dieser Kurs in die fortgeschrittene parallele Programmierung für gemeinsamen Speicher mit Threads ein.  Aufbauend auf den Inhalten der Vorlesung Multithreading in C++ werden die folgenden Themen behandelt: <ul style="list-style-type: none"><li>• C++ Speichermodell und atomare Operationen</li><li>• Entwurf lockfreier nebenläufiger Datenstrukturen</li><li>• Fortgeschrittenes Thread-Management (z.B. Thread Pools)</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung besucht haben, haben Sie erweiterte Kompetenz in der Entwicklung paralleler Programme und sind in der Lage <ul style="list-style-type: none"><li>- Systematisch korrekte und effiziente parallele Programme zu entwickeln</li><li>- Parallele Datenstrukturen zu entwerfen und umzusetzen</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: <ul style="list-style-type: none"><li>• Kenntnisse in C/C++</li><li>• Basiskenntnisse der Programmierung von Threads in C++ (lockbasierte Synchronisation und lockbasierte nebenläufige Datenstrukturen)</li></ul>				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0977-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0977-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Skalierbare Datenmanagement-Systeme					
Modul Nr. 20-00-1017	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1017-iv	Skalierbares Datenmanagement	6	Integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Diese Vorlesungen ist eine Einführung in die Basiskonzepte und die wesentlichen Paradigmen für skalierbare Datenmanagement-Systeme. Der Fokus der Vorlesung ist auf die systemorientieren Aspekten und Interna solcher Systeme gerichtet, um große Datenmengen zu speichern, zu ändern, und zu analysieren.  Themen der Vorlesung sind:  Database Architectures Parallel and Distributed Databases Data Warehousing MapReduce and Hadoop Spark and its Ecosystem Optional: NoSQL Databases, Stream Processing, Graph Databases, Scalable Machine Learning				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem Kurs haben die Studierenden einen Überblick über die wichtigsten Konzepte, Algorithmen und System-Aspekte für skalierbare Datenmanagement-Systeme erworben. Das Hauptziel ist es, dass die Studierenden das Wissen besitzen, solche Systeme zu designen und zu entwickeln, inklusive praktischer Übungen auf Basis von bestehenden Systemen wie Spark.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Programmierkenntnisse in C++ and Java Der vorherige Besuch von Informationsmanagement oder einer vergleichbaren Veranstaltung				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1017-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1017-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Advanced Data Management Systems					
<b>Modul Nr.</b> 20-00-1039	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1039-iv	Advanced Data Management Systems	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> <p>Dies ist eine fortgeschrittene Veranstaltung aus dem Bereich der Architektur und Implementierung moderner Datenbanksysteme mit dem speziellen Fokus auf System-orientieren Aspekten und Interna solcher Systeme. Mögliche Themengebiete die in der Vorlesung behandelt werden sind: moderne Hardwaretechnologien für das Datenbanksysteme, Optimierungen für Hauptspeicherdatenbanken, Parallelisierungsstrategien und Approximative Anfrageausführung usw.</p> <p>Es wird erwartet, dass für jede Vorlesung aktuelle Veröffentlichungen (SIGMOD, VLDB, etc.) vorher gelesen werden. Die Hauptideen ausgewählter Veröffentlichungen werden in Programmierprojekten umgesetzt.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach erfolgreichem Abschluss der Veranstaltung haben Studierende ein vertieftes Verständnis von aktuellen Techniken für das Design von modernen Datenbanksystemen. Die Studierenden können Vor- und Nachteile dieser Techniken mit dem Fokus auf möglichen Verbesserungen diskutieren. Sie können einzelne Techniken implementieren und experimentelle Evaluierungen dieser Techniken zum Vergleich von Designalternativen durchführen.</p>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Solide Programmierkenntnisse in C and C++ Der vorherige Besuch von „Skalierbare Datenmanagement-Systeme“ und „Informationsmanagement“ oder vergleichbaren Veranstaltungen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1039-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1039-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>



## Modulbeschreibung

<b>Modulname</b>					
Advanced C++ modern programming					
<b>Modul Nr.</b> 20-00-1068	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1068-iv	Advanced C++ modern programming	3	Integrierte Veranstaltung	2
<b>2</b>	<b>Lerninhalt</b> Die Vorlesung wird die letzten Änderungen und Erweiterungen der Sprache C++ behandeln und insbesondere auf die Standards: ISO/IEC 14882:2011, 14882:2014, and 14882:2017 eingehen.  Die Liste der Themen: 1. Einführung in modernes C++ 2. Verbessertes Typsystem 3. Uniforme Initialisierung 4. Moderner Ansatz in Hinblick auf den Entwurf und Implementierung von Klassen 5. Verbesserungen für die Entwicklung von Bibliotheken 6. Moderne "generische Programmierung" 7. Einführung in die Metaprogrammierung 8. Vereinfachung von Code durch den Einsatz von Standardkomponenten 9. STL: Containers, Algorithmen und Iteratoren 10. Neueste Entwicklungen: C++17 11. Die Zukunft von C++: C++20				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> + Die Studierenden werden in der Lage sein die Hauptunterschiede zwischen den modernen C++ Standards zu benennen + Die Studierenden haben ein vertieftes Verständnis moderner "generischer Programmierung" + Die Studierenden sind in der Lage die neuen Hauptkomponenten der C++ Standardbibliothek zu verwenden + Die Studierenden können Abwägungen zwischen Flexibilität und Performance in modernen C++ nachvollziehen + Die Studierenden haben ein Überblick über die Wahrscheinlichsten Entwicklungsschritte				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:				

	+ Grundlagenwissen von C und C++ + Vertrautheit mit den Grundlagen objektorientierter und generischer Programmierung + Grundlagenwissen im Bereich funktionale Programmierung
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1068-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1068-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Konzepte der Programmiersprachen					
Modul Nr. 20-00-1117	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1117-iv	Konzepte der Programmiersprachen	6	Integrierte Veranstaltung	4
2	Lerninhalt Kurze Einführung und Geschichte der Programmiersprachen, Kriterien zur Messung von Programmiersprachen, Grundkonzepte der PL wie Syntax, Semantik, Variablen, Namen, Bindungen, Umfang, Subprogram, Expressionen, Arrays, Pointers, abstrakte Typen, funktionale Programme				
3	Qualifikationsziele / Lernergebnisse Die Studierenden werden am Ende des Kurses in der Lage sein, die zugrundeliegenden Mechanismen der wichtigsten Konzepte hinter Programmiersprachen zu verstehen. Die Studierenden werden auch Erfahrung erhalten, eine einfache Programmiersprache mit einer beliebigen Sprache Workbench namens MPS als Gruppenprojekt zu bauen.				
4	Voraussetzung für die Teilnahme Keine				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1117-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%).				
7	Benotung Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-1117-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Data Science and Engineering**

**Wahlbereich Foundations of Data Science**

## Modulbeschreibung

<b>Modulname</b> Data Mining und Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0052	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0052-iv	Data Mining und Maschinelles Lernen	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b>				
	Durch die rasante Entwicklung der Informationstechnologie sind immer größere Datenmengen verfügbar. Diese enthalten oft implizites Wissen, das, wenn es bekannt wäre, große wirtschaftliche oder wissenschaftliche Bedeutung hätte. Data Mining ist ein Forschungsgebiet, das sich mit der Suche nach potentiell nützlichem Wissen in großen Datenmengen beschäftigt, und Maschinelles Lernverfahren gehören zu den Schlüsseltechnologien innerhalb dieses Gebiets.				
	Die Vorlesung bietet eine Einführung in das Gebiet des Maschinellen Lernens unter dem besonderen Aspekt des Data Minings. Es werden Verfahren aus verschiedenen Paradigmen des Maschinellen Lernens mit exemplarischen Anwendungen vorgestellt. Um das Wissen zu operationalisieren, werden in den Übungen prak-tisch-e Erfahrungen mit Lernalgorithmen gesammelt.				
	• Einführung (Grundbegriffe, Lernprobleme, Konzepte, Beispiele, Repräsentation)				
	• Regel-Lernen				
	◦ Lernen einzelner Regeln (Generalisierung und Spezialisierung, Strukturierte Hypothesenräume, Version Spaces)				
	◦ Lernen von Regel-Mengen (Covering Strategie, Evaluierungsmaße für Regeln, Pruning, Mehr-Klassenprobleme)				
	• Evaluierung und kosten-sensitives Lernen (Accuracy,X-Val,ROC-Kurven,Cost-Sensitive Learning)				
	• Instanzenbasiertes Lernen (kNN,IBL,NEAR,RISE)				
	• Entscheidungsbaum-Lernen (ID3, C4.5, etc.)				
	• Ensemble-Methoden (Bias/Variance, Bagging, Randomization, Boosting, Stacking, ECOCs)				
	• Pre-Processing (Feature Subset Selection, Diskretisierung, Sampling, Data Cleaning)				
• Clustering und Lernen von Assoziationsregeln (Apriori)					

3	<p><b>Qualifikationsziele / Lernergebnisse</b></p> <p>Nach der erfolgreichen Absolvierung dieser Lehrveranstaltung sind die Studierenden in der Lage</p> <ul style="list-style-type: none"> <li>• grundlegende Techniken des Data Mining und Maschinellen Lernens zu verstehen und erklären</li> <li>• praktische Data Mining Systeme selbständig einsetzen und deren Stärken und Schwächen verstehen</li> <li>• neue Entwicklungen auf diesem Gebiet kritisch beurteilen</li> </ul>
4	<p><b>Voraussetzung für die Teilnahme</b></p>
5	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0052-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0052-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning  M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>

9	<b>Literatur</b> <ul style="list-style-type: none"> <li>• Mitchell: Machine Learning, McGraw-Hill, 1997</li> <li>• Ian H. Witten and Eibe Frank: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan-Kaufmann, 1999</li> </ul>
10	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b> Statistisches Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0358	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0358-iv	Statistisches Maschinelles Lernen	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> - Statistische Methodik für das Maschinelle Lernen - Auffrischung zu Statistik, Optimierung und Linearer Algebra - Bayes'sche Entscheidungstheorie - Wahrscheinlichkeitsdichtenschätzung - Nichtparametrische Modelle - Mixtur Modelle und der EM-Algorithmus - Lineare Modelle zur Klassifikation und Regression - Statistische Lerntheorie - Kernel Methoden zur Klassifikation und Regression				
3	<b>Qualifikationsziele / Lernergebnisse</b> Die Lehrveranstaltung ist eine systematische Einführung in die Grundlagen und Methodik des statistischen maschinellen Lernens. Nach erfolgreichen Abschluss der Lehrveranstaltung, verstehen Studierende die wichtigsten Methoden und Ansätze des Statistischen Maschinellen Lernens. Sie können maschinelle Lernverfahren anwenden, um eine Vielzahl neuer Probleme zu lösen.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0358-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0358-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> <ol style="list-style-type: none"> <li>1. C.M. Bishop, Pattern Recognition and Machine Learning (2006), Springer</li> <li>2. K.P. Murphy, Machine Learning: a Probabilistic Perspective (expected 2012), MIT Press</li> <li>3. D. Barber, Bayesian Reasoning and Machine Learning (2012), Cambridge University Press</li> <li>4. T. Hastie, R. Tibshirani, and J. Friedman (2003), The Elements of Statistical Learning, Springer Verlag</li> <li>5. D. MacKay, Information Theory, Inference, and Learning Algorithms (2003), Cambridge University Press</li> <li>6. R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification (2nd ed. 2001), Wiley-Interscience</li> <li>7. T.M. Mitchell, Machine Learning (1997), McGraw-Hill</li> </ol>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Statistical Relational Artificial Intelligence: Logic, Probability, and Computation					
<b>Modul Nr.</b> 20-00-1011	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1011-iv	Statistical Relational Artificial Intelligence: Logic, Probability, and Computation	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> + logische Programmierung + Lernen von logischen Programmen aus Daten + Probabilistische Graphische Modelle: Inferenz und Lernen + Statistisch-Relationale Modelle wie z.B. ProbLog und Markov Logic Networks + Schlussfolgern in statistisch-relationalen Modellen + Lernen von statistisch-relationalen Modellen aus Daten + Relationale lineare und quadratische Programme				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Lehrveranstaltung ist eine systematische Einführung in die Grundlagen und Methodik des statistisch-relationalen Lernens und Künstlichen Intelligenz: Das Studium und Design von intelligenten Agenten, die in verrauschten Welten agieren, die aus Individuen (Objekte, Dinge) und komplexe Beziehungen zwischen den Individuen bestehen. Nach erfolgreichem Abschluss der Lehrveranstaltung verstehen Studierende die wichtigsten Methoden und Ansätze in der statistisch-relationalen Künstlichen Intelligenz. Sie verstehen die grundlegenden Herausforderungen von relationalen Domänen. Sie kennen aktuelle Ansätze, um diese Herausforderungen zu lösen. Sie sind außerdem in der Lage ihre Kenntnisse auf aktuelle Probleme anzuwenden.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Der vorherige Besuch von “Statistisches Maschinelles Lernen” und “Probabilistische Graphische Modelle” oder vergleichbarer Veranstaltungen ist empfohlen, ist aber keine Voraussetzung.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1011-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1011-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden regelmäßig aktualisiert und beinhalten beispielsweise:</p> <p>Luc De Raedt, Kristian Kersting, Sriraam Natarajan, David Poole (2016): Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan &amp; Claypool Publishers, ISBN: 9781627058414.</p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Tiefe Generative Modelle					
Modul Nr. 20-00-1035	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1035-iv	Tiefe Generative Modelle	6	Integrierte Veranstaltung	4
2	Lerninhalt Generative Modelle, implizite und explizite Modelle, Variational AutoEncoders, Generative Adversarial Networks, Numerische Optimierung für generative Modelle, Anwendungen in der medizinischen Bildverarbeitung				
3	Qualifikationsziele / Lernergebnisse Nachdem Studierende das Modul besucht haben, können sie - den Aufbau und die Funktionsweise Tiefer Generativer Modelle (Deep Generative Models, DGM) erklären - wissenschaftliche Veröffentlichungen zum Thema DGMs kritisch hinterfragen und damit fachlich beurteilen - grundlegende DGMs in einer dafür ausgelegten höheren Programmiersprache selbstständig konstruieren / implementieren - die Implementierung und Anwendung von DGMs auf unterschiedliche Anwendungen übertragen				
4	Voraussetzung für die Teilnahme Empfohlen:  - Programmierkenntnisse Python - Kenntnisse aus dem Bereich Lineare Algebra - Der vorherige Besuche von „Bildverarbeitung“, „Computer Vision I“ und „Statistisches Maschinelles Lernen“ oder vergleichbarer Veranstaltungen				
5	Prüfungsform Bausteinbegleitende Prüfung:  • [20-00-1035-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1035-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Wird in Veranstaltung bekannt gegeben.
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Model Checking					
Modul Nr. 20-00-1115	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1115-vl	Model Checking	3	Vorlesung	2
2	Lerninhalt * Temporallogiken: - Lineare temporal Logik (LTL), Computation Tree Logic (CTL) und CTL*: Syntax, Semantik, Komplexität * Modelprüfungsverfahren für LTL, CTL, CTL*, insbesondere Büchiautomaten * Partial Order Reduction * Timed Automata				
3	Qualifikationsziele / Lernergebnisse Im Rahmen des Kurses sollen die Studierenden folgende Fähigkeiten erwerben: * Verständnis der theoretischen Grundlagen der Temporallogiken LTL, CTL und CTL* * Fähigkeit zur Auswahl der geeigneten Logik zur Spezifikation und Modellprüfung in Abhängigkeit von dem zu modellierenden System und der zu prüfenden Eigenschaft * Verfahren und Techniken zur Modellprüfung (Model Checking) z.B. Modellprüfung mit Büchiautomaten, Partial Order Reduction u.ä. * Wissen über die Charakteristika und Grenzen der Modellprüfung * Kenntnisse in der Modellprüfung von Timed Automate * Fähigkeit zur Anwendung von Tools zur Modellprüfung				
4	Voraussetzung für die Teilnahme Empfohlen werden Kenntnisse in * Aussagenlogik * Deduktionssystemen * Automatentheorie				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1115-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%).</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1115-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>



**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Data Science and Engineering**

**Wahlbereich Studienbegleitende Leistungen**

**Praktika, Projektpraktika und ähnliche  
Veranstaltungen**

## Modulbeschreibung

<b>Modulname</b> Implementierung von Programmiersprachen					
<b>Modul Nr.</b> 20-00-0306	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0306-pr	Implementierung von Programmiersprachen	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Es werden Konzepte der Implementierung von Programmiersprachen vermittelt. Ferner werden diese Konzepte angewendet, um Erweiterungen für Programmiersprachen zu implementieren.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Fähigkeit, eine professionelle Aufgabe aus der Informatik selbstständig und erfolgreich nach den anerkannten Grundsätzen der Profession zu bearbeiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Es wird kein Vorwissen vorausgesetzt. Jedoch sind gute Programmiererfahrungen sowie Kenntnisse über Kompilerverbau und virtuelle Maschinen von Vorteil.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0306-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:  • [20-00-0306-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Software Development Tools					
<b>Modul Nr.</b> 20-00-0673	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0673-pr	Software Development Tools	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Die Entwicklung von Werkzeugen zur Unterstützung der Entwicklung von Software.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Gewinnen von praktischer Erfahrung in der Entwicklung von Softwareentwicklungswerkzeugen. Verstehen der Grenzen von Softwareentwicklungswerkzeugen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch der Veranstaltung „Software Engineering“ oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0673-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0673-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				
<b>8</b>	<b>Verwendbarkeit des Moduls</b>				

	B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Forschungsprojekt Knowledge Engineering und Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0751	<b>Leistungspunkte</b> 12 CP	<b>Arbeitsaufwand</b> 360 h	<b>Selbststudium</b> 240 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0751-pj	Forschungsprojekt Knowledge Engineering und Maschinelles Lernen	12	Projekt	8
2	<b>Lerninhalt</b> An einem individuellen Projekt soll das eigenständige Forschen in den Gebieten Knowledge Engineering, Künstliche Intelligenz, maschinelles Lernen und Data Mining unter Anleitung erlernt werden. Dabei werden die Themen jeweils in Zusammenarbeit mit dem Betreuer definiert. Mögliche Themenfelder: <ul style="list-style-type: none"><li>• Maschinelles Lernen und Data Mining</li><li>• Induktives Regel-Lernen</li><li>• Learning from Preferences</li><li>• Multilabel Classification</li><li>• Information Extraction</li><li>• Web Mining</li><li>• Semantic Web</li><li>• Game Playing</li></ul> Konkrete Aufgabenstellungen werden individuell vereinbart, und das Projekt kann jederzeit begonnen werden. Studierende, die an einem derartigen Projekt interessiert sind, wenden sich bitte an einen Mitarbeiter des anbietenden Fachgebiets.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach Bearbeitung dieses Projekts sind die Studierenden in der Lage <ul style="list-style-type: none"><li>• selbständig kleinere Forschungsarbeiten in den Bereichen Knowledge Engineering, Künstliche Intelligenz, maschinelles Lernen und Data Mining durchzuführen</li><li>• diese Ergebnisse in einem Abschlußreport zu dokumentieren</li><li>• in einem wissenschaftlichen Vortrag zu präsentieren</li></ul>				

	<ul style="list-style-type: none"> <li>• in einer kritischen Diskussion zu verteidigen</li> </ul>
4	<p><b>Voraussetzung für die Teilnahme</b></p> <p>Empfohlen: Grundlegende Kenntnisse in den Bereichen Knowledge Engineering, Künstliche Intelligenz, Data Mining und maschinelles Lernen sind hilfreich. Es werden außerdem grundlegende Kenntnisse in einer Programmiersprache (z.B. Java o.ä.) vorausgesetzt.</p> <p>Darüber hinaus ist aber besonders die Motivation zur selbstständigen Arbeit und das Interesse an aktuellen Forschungsfragen relevant.</p>
5	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0751-pj] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0751-pj] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Lernende Roboter: Integriertes Projekt, Teil 1					
<b>Modul Nr.</b> 20-00-0753	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0753-pj	Lernende Roboter: Integriertes Projekt, Teil 1	6	Praktikum	4
2	<b>Lerninhalt</b> In "Lernende Roboter: Integriertes Projekt, Teil 1" wird zunächst von Studierenden unter Anleitung eine aktuelle Problemstellung des Roboter-Lernens erarbeitet, welche den Forschungsinteressen der Studierenden entspricht, und eine Literaturstudie durchgeführt. Basierend auf diesen Vorarbeiten werden ein Projektplan ausgearbeitet, die notwendigen Algorithmen erprobt und eine prototypische Realisierung in Simulation erstellt.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichen Abschluss der Lehrveranstaltung, können Studierende unabhängig kleine Forschungsprojekte im Bereich Robot Learning aufbauen und in Simulation erproben.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Gleichzeitiger oder vorheriger Besuch der Vorlesung „Lernende Roboter“ oder einer vergleichbaren Veranstaltung.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0753-pj] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung:				



	<ul style="list-style-type: none"> <li>• [20-00-0753-pj] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Lernende Roboter: Integriertes Projekt, Teil 2					
<b>Modul Nr.</b> 20-00-0754	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0754-pj	Lernende Roboter: Integriertes Projekt, Teil 2	6	Praktikum	4
2	<b>Lerninhalt</b> In "Lernende Roboter: Integriertes Projekt, Teil 2" werden die Lösungen aus dem "Teil 1" vervollständigt und auf einen realen Roboter angewandt. Ein wissenschaftlicher Artikel wird über die Fragestellung, Methoden und Ergebnisse geschrieben sowie ggf. eingereicht.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Abschluss der Lehrveranstaltung können Studierende unabhängig kleine Forschungsprojekte im Bereich Robot Learning aufbauen und in Simulation erproben.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Gleichzeitiger oder vorheriger Besuch der Vorlesung „Lernende Roboter“ oder einer vergleichbaren Veranstaltung.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0754-pj] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-0754-pj] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Projekt-Praktikum Knowledge Engineering und Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0919	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0919-pp	Projekt-Praktikum Knowledge Engineering und Maschinelles Lernen	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> Im Rahmen des Projektpraktikums implementieren Studierende eine vordefinierte, größere Aufgabe aus den Gebieten Knowledge Engineering, Künstliche Intelligenz, maschinelles Lernen und Data Mining. Dabei werden die Themen jeweils in Zusammenarbeit mit dem Betreuer definiert.  Mögliche Themenfelder: - Maschinelles Lernen und Data Mining - Induktives Regel-Lernen - Learning from Preferences - Multilabel Classification - Information Extraction - Web Mining - Semantic Web - Game Playing  Konkrete Aufgabenstellungen werden individuell vereinbart, und das Praktikum kann jederzeit begonnen werden.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Bearbeitung dieses Projekts sind die Studierenden in der Lage - selbständig größere Programmieraufgaben in den Bereichen Knowledge Engineering, Künstliche Intelligenz, Maschinelles Lernen und Data Mining durchzuführen - mit Hilfe der implementierte Instrumente wissenschaftliche Experimente und Evaluierungen durchzuführen				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlegende Kenntnisse in den Bereichen Knowledge Engineering, Künstliche Intelligenz, Data Mining und maschinelles Lernen sind hilfreich. Es werden außerdem grundlegende Kenntnisse in einer Programmiersprache (z.B. Java o.ä.) vorausgesetzt.				

	Darüber hinaus ist aber besonders die Motivation zur selbstständigen Arbeit und das Interesse an aktuellen Forschungsfragen relevant.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0919-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. <ul style="list-style-type: none"> <li>Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)</li> </ul>
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0919-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Softwareprojekt Datenanalyse für natürliche Sprache					
<b>Modul Nr.</b> 20-00-0948	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0948-pp	Softwareprojekt Datenanalyse für natürliche Sprache	9	Praktikum	6
<b>2</b>	<b>Lerninhalt</b> Große Datenmengen sind heute eine wertvolle Informationsquelle. Allerdings ist nur durch die Verwendung von intelligenter Datenanalyse das volle Potential dieser Daten nutzbar. Solche Methoden ermöglichen es neue und praktisch nutzbare Informationen in großen natürlichsprachlichen Daten zu identifizieren und unterstützen dadurch die Entscheidungsfindung bei komplexen Aufgaben. In diesem Projekt werden Studenten eigene Ideen und neue Softwaresysteme entwickeln die es ermöglichen Informationen für verschiedene Aufgaben aus einer großen Menge natürlichsprachlicher Texte (Big Data) zu extrahieren. Das jeweilige Rahmenthema der Veranstaltung wechselt jedes Semester und wird auf der Fachgebietshomepage bekannt gegeben.  Weitere Informationen: <a href="https://www.ukp.tu-darmstadt.de/teaching/courses/software-project/">https://www.ukp.tu-darmstadt.de/teaching/courses/software-project/</a>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung besucht haben, können sie - sprachtechnologische Frameworks verstehen und einsetzen, - komplexe NLP-Systeme eigenständig planen und umsetzen, - große natürlichsprachliche Daten analysieren und - die eigenen Ergebnisse mündlich und schriftlich präsentieren.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  - Programmierkenntnisse (Scala, Java oder Python) - Interesse mit Texten aus natürlicher Sprache zu arbeiten				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0948-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0948-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Projektpraktikum Deep Learning in der Computer Vision					
<b>Modul Nr.</b> 20-00-0980	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0980-pp	Projektpraktikum Deep Learning in der Computer Vision	9	Praktikum	6
<b>2</b>	<b>Lerninhalt</b> Im Rahmen des Projektpraktikums werden ausgewählte Themen aus dem Bereich des Deep Learning (tiefe neuronale Netze) für Fragestellungen in der Computer Vision in Gruppen bearbeitet. Dazu gehört die praktische Umsetzung mit modernen Deep Learning Frameworks. Die Ergebnisse werden am Ende in einem Vortrag vorgestellt. Die konkreten Themen orientieren sich am aktuellen Stand der Forschung und wechseln von Semester zu Semester.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Durch erfolgreiche Teilnahme erwerben Studierende vertiefte Kenntnisse in tiefen neuronalen Netzen und deren Anwendungen in der Computer Vision. Sie können aktuelle Techniken in diesem Bereich analysieren, modifizieren und anwenden. Sie trainieren weiterhin Präsentationsfähigkeiten und die Arbeit in einem Team.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  * Gute Programmierkenntnisse in C/C++ oder Python oder Lua  * Vorherige oder parallele Belegung von "Computer Vision I" oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0980-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				



<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0980-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Parallele Programmiertechnologie					
<b>Modul Nr.</b> 20-00-1008	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1008-pr	Parallele Programmiertechnologie	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Die Praktikumsteilnehmer*innen entwickeln Technologien zur parallelen Programmierung aus den folgenden Themenbereichen und/oder wenden diese an: <ul style="list-style-type: none"><li>• Erschließung möglicher Parallelität</li><li>• Leistungsanalyse und –modellierung</li><li>• Korrektheitsanalyse</li><li>• Profiling</li><li>• Skalierbare Algorithmen</li><li>• Ressourcenmanagement und Scheduling</li><li>• Anwendungen (z.B. Deep Learning)</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <ul style="list-style-type: none"><li>• Kennenlernen und praktische Entwicklung und/oder Anwendung paralleler Programmiertechnologien</li><li>• Einüben softwaretechnischer Methoden</li><li>• Teamarbeit in Softwareprojekten</li><li>• Präsentation von Projektergebnissen in Berichten und Vorträgen</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Kenntnisse paralleler Programmierung und Systeme				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-1008-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1008-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Data Management - Praktikum					
<b>Modul Nr.</b> 20-00-1041	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1041-pr	Data Management - Praktikum	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Die Teilnehmenden lösen in kleinen Projektgruppen ein gegebenes Problem. Bei den Problemen handelt es sich um Programmierprojekte, die sich auf Fragestellungen aus aktuellen Forschungsthemen des Data Management Lab beziehen.  Mögliche Themenbereiche sind: - Skalierbare Datenbanksysteme und moderne Hardware - Cloud Datenbanken und Blockchains - Interaktive Daten- und Textexploration - Natural Language Interfaces für Datenbanken - Skalierbare Systeme für Maschinelles Lernen  In dieser Veranstaltung setzen Studierende ein ausgewähltes Projekt um. Im Vergleich zum Praktikum haben die Probleme des Projektpraktikums einen erweiterten Umfang.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Beendigung der Veranstaltung haben Studierende folgende Lernziele erreicht: - Vertieftes Verständnis von aktuellen Techniken für moderne Datenmanagement-Systeme - Anwendung und Implementierung der Techniken in individuellen Projekten - Evaluierung von möglichen Designalternativen mit Hilfe von Benchmarks bzw. realen Workloads				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Abhängig vom ausgewählten Thema.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1041-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1041-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Data Management - Projektpraktikum					
<b>Modul Nr.</b> 20-00-1042	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1042-pp	Data Management - Projektpraktikum	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> Die Teilnehmenden lösen in kleinen Projektgruppen ein gegebenes Problem. Bei den Problemen handelt es sich um Programmierprojekte, die sich auf Fragestellungen aus aktuellen Forschungsthemen des Data Management Lab beziehen.  Mögliche Themenbereiche sind: - Skalierbare Datenbanksysteme und moderne Hardware - Cloud Datenbanken und Blockchains - Interaktive Daten- und Textexploration - Natural Language Interfaces für Datenbanken - Skalierbare Systeme für Maschinelles Lernen  In dieser Veranstaltung setzen Studierende ein ausgewähltes Projekt um. Im Vergleich zum Praktikum haben die Probleme des Projektpraktikums einen erweiterten Umfang.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Beendigung der Veranstaltung haben Studierende folgende Lernziele erreicht: - Vertieftes Verständnis von aktuellen Techniken für moderne Datenmanagement-Systeme - Anwendung und Implementierung der Techniken in individuellen Projekten - Evaluierung von möglichen Designalternativen mit Hilfe von Benchmarks bzw. realen Workloads				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Abhängig vom ausgewählten Thema.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1042-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1042-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Anwendung von Reinforcement Learning Methoden					
<b>Modul Nr.</b> 20-00-1048	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1048-pp	Anwendung von Reinforcement Learning Methoden	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> In diesem Projekt lernen Studierende das experimentelle Arbeiten in einem interdisziplinären Team, und bekommen so Einblicke in das wissenschaftliche Arbeiten im Reinforcement Learning. Im Projekt entwickeln Kleingruppen unter Anleitung ein gemeinsames Experiment im Reinforcement Learning basierend auf speziellen Plattformen (Cartpole, Furuta-Pendel, etc), werten dieses aus und schreiben einen Forschungsbericht/Paper.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b>  Praktische Einführung in das wissenschaftliche Arbeiten, Durchführung eines Experimentes von der Forschungsidee bis hin zur Veröffentlichung.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Gleichzeitige oder vorhergehende Belegung der Vorlesung "Reinforcement Learning: Von Grundlagen zu den Tiefen Ansätzen" oder "Lernende Roboter" oder vergleichbarer Veranstaltungen.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1048-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>  Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b>				



	Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1048-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Expertenpraktikum im Robot Learning					
<b>Modul Nr.</b> 20-00-1108	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1108-pp	Expertenpraktikum im Robot Learning	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> In diesem Projekt perfektionieren Studierende das experimentelle Arbeiten in einem interdisziplinären Team, und entwickeln sich zu Experten im wissenschaftliche Arbeiten im Robot Learning. Im Projekt entwickeln in einer Kleingruppen unter Anleitung ein gemeinsames Experiment im Robot Learning basierend auf speziellen Robotik-Plattformen, werten dieses aus und schreiben einen Forschungsbericht/Paper, welches die Qualität einer Einreichung bei einer internationalen wissenschaftlichen Konferenz oder Zeitschrift erreicht.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung besucht haben, können sie die praktischen Fertigkeiten eines Experten im wissenschaftlichen Arbeiten im Roboter Lernen anwenden. Sie sind in der Lage, Experimente von der Forschungsidee bis hin zur Veröffentlichung zu analysieren und synthetisieren				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen wird die erfolgreiche Durchführung von Lernende Roboter: Integriertes Projekt - Teil 1 und Lernende Roboter: Integriertes Projekt - Teil 2				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1108-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1108-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Leistungsanalyse und Modellierung von Softwaresystemen					
<b>Modul Nr.</b> 20-00-1130	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1130-pr	Leistungsanalyse und Modellierung von Softwaresystemen	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> <p>Ziel dieses Praktikums ist:</p> <ol style="list-style-type: none"><li>1) Erfahrungen im Design von Experimenten zu sammeln, die die Leistung komplexer Softwaresysteme messen,</li><li>2) Ergebnisse verarbeiten und analysieren</li><li>3) Modelle erstellen, die das Verhalten des Systems beschreiben.</li></ol> <p>Die experimentellen Ergebnisse und die Modelle werden verwendet, um die Komponenten des Systems zu bestimmen, die den Engpass für die Leistung darstellen.</p> <p>Die im Praktikum erworbenen Fähigkeiten sollen für eine Vielzahl von Karrierewegen relevant sein: Studierende, die später in Systeme-Bereich promovieren, Data Scientists, die mit großen verteilten Pipelines arbeiten werden, Software Engineers und DevOps, die an der Verbesserung der Leistung von IT-Systemen arbeiten.</p> <p>Um die oben genannten Ziele zu erreichen, werden wir einerseits über die relevante Theorie sprechen (z.B., Statistical methods, Little's Law, Queuing Theory) und andererseits eine Datenverarbeitungsanwendung implementieren. Anschließend wird die Anwendung einem Benchmarking unterzogen und detailliert modelliert.</p> <p>Das Praktikum schließt mit kurzen Projektpräsentationen ab, in denen die Studierenden zeigen, dass sie das Verhalten ihrer Implementierung verstanden haben und anhand der experimentellen Daten und der von ihnen erstellten Modelle Ideen zur Beseitigung von Bottlenecks liefern können</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach Besuch der Veranstaltung können Studierende:</p> <ul style="list-style-type: none"><li>* Zwischen verschiedenen Experimenttypen wählen, um leistungsbezogene Fragen zu einem Softwaresystem zu beantworten</li><li>* Detaillierte Modelle eines Softwaresystems erstellen</li><li>* Bottleneck-Analysen durchführen</li></ul>				

	* die Ergebnisse der Experimente zusammenfassen und präsentieren
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Die Studierenden sollten Einführungskurse in Computerarchitektur, Betriebssysteme und Java-Programmierung besucht haben.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1130-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1130-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Projektseminar Autonomes Fahren I					
Modul Nr. 18-su-2070	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 135 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. rer. nat. Andreas Schürr		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	18-su-2070-pj	Projektseminar Autonomes Fahren I	0	Projektseminar	3
2	Lerninhalt				
	<ul style="list-style-type: none"><li>• Praktische Programmiererfahrung mit C++ bei der Entwicklung eingebetteter Systemsoftware aus dem Bereich des autonomen Fahrens anhand eines Modellautos</li><li>• Anwenden von Regelungs- und Steuerungsmethoden aus dem Bereich des autonomen Fahrens</li><li>• Einsatz von Software-Engineering-Techniken (Design, Dokumentation, Test, ...) eines nicht trivialen eingebetteten Software-Systems mit harten Echtzeit-Anforderungen und beschränkten Ressourcen (Speicher, ...)</li><li>• Nutzung eines vorgegebenen Software-Rahmenwerks und Anwendung von weiteren Bibliotheken inklusive eines modular aufgebauten (Echtzeit-)Betriebssystems</li><li>• Einsatz von Source-Code-Management-Systemen, Zeiterfassungswerkzeugen und sonstigen Projektmanagement-Tools</li><li>• Präsentation von Projektergebnissen im Rahmen von Vorträgen</li></ul>				
3	Qualifikationsziele / Lernergebnisse Studierende sammeln im Rahmen dieses Moduls praktische Erfahrung in der Software-Entwicklung für eingebettete Systeme aus dem Bereich des autonomen Fahrens anhand eines Modellautos. Dabei lernen sie in Teamarbeit eine umfangreiche Aufgabe zu bewältigen. Zur Lösung dieser Aufgabe wird geübt, dass in der Gruppe vorhandene theoretische Wissen (aus anderen Lehrveranstaltungen wie Echtzeitsysteme, Software-Engineering - Einführung, C++ Praktikum, Digitale Regelungssysteme) gezielt zur Lösung der praktischen Aufgabe einzusetzen.				

	<p>Studierende, die an diesem Modul erfolgreich teilgenommen haben, sind in der Lage, zu einer vorgegebenen Problemstellung ein größeres Softwareprojekt in einem interdisziplinären Team eigenständig zu organisieren und auszuführen. Die Teilnehmer erwerben folgende Fähigkeiten im Detail:</p> <ul style="list-style-type: none"> <li>• Eigenständiges Einarbeiten in ein vorgegebenes Rahmenwerk und vorgefertigten Bibliotheken</li> <li>• Umsetzung von theoretischem Wissen in ein Softwaresystem</li> <li>• Umfangreicher Einsatz von Werkzeugen zur Versions-, Konfiguration- und Änderungsverwaltung</li> <li>• Realistische Zeitplanung und Ressourceneinteilung (Projektmanagement)</li> <li>• Entwicklung von Hardware-/Software-Systemen mit C++ unter Berücksichtigung wichtiger Einschränkungen eingebetteter Systeme</li> <li>• Planung und Durchführung umfangreicherer Qualitätssicherungsmaßnahmen</li> <li>• Zusammenarbeit und Kommunikation in und zwischen mehreren Teams</li> </ul>
4	<p><b>Voraussetzung für die Teilnahme</b></p> <ul style="list-style-type: none"> <li>• ETiT, WI-ETiT (DT), iST, Informatik: Grundlegende Softwaretechnik-Kenntnisse sowie vertiefte Kenntnisse objektorientierter Programmiersprachen (insbesondere: C++)</li> </ul> <p>Zusätzlich erwünscht:</p> <ul style="list-style-type: none"> <li>• Grundlagen der Entwicklung von Echtzeitsystemen oder der Bildverarbeitung</li> <li>• ETiT, WI-ETiT (AUT), MEC: Grundlagen der Regelungstechnik, Reglerentwurf im Zustandsraum, ggf. Grundlagen der digitalen Regelung</li> </ul>
5	<p><b>Prüfungsform</b> Modulabschlussprüfung:</p> <ul style="list-style-type: none"> <li>• Modulprüfung (Studienleistung, mündliche Prüfung, Dauer 30 Min, Standard)</li> </ul>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p>
7	<p><b>Benotung</b> Modulabschlussprüfung:</p> <ul style="list-style-type: none"> <li>• Modulprüfung (Studienleistung, mündliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p>

	MSc ETiT, BSc iST
<b>9</b>	<b>Literatur</b> <a href="https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/ps-af-i/">https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/ps-af-i/</a> und Moodle
<b>10</b>	<b>Kommentar</b>



**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Data Science and Engineering**

**Wahlbereich Studienbegleitende Leistungen**

**Seminare**

## Modulbeschreibung

<b>Modulname</b>					
Software Engineering - Projektseminar					
<b>Modul Nr.</b> 20-00-0359	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0359-se	Software Engineering - Projekt Seminar	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b>				
	<ul style="list-style-type: none"><li>• Angebotsmesse der Auftraggeber</li><li>• Projektauswahl</li><li>• Anforderungsanalyse beim externen Auftraggeber</li><li>• Präsentation des Pflichtenheftes insbesondere der Projektorganisation und des iterativen Entwicklungsplans</li><li>• Analyse der Werkzeuge und der Designkonzepte</li><li>• Präsentation der Architektur und des Designs risikobehafteter Funktionen</li><li>• Design und Implementierung der Iterationen</li><li>• Präsentation der Implementierung und der Qualitätssicherung</li><li>• Präsentation des abgeschlossenen Projekts der nächsten Studentengeneration</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b>				
	<ul style="list-style-type: none"><li>• Erfahrung mit selbständiger Durchführung von Softwareprojekten mittleren Umfangs</li><li>• Fähigkeit die verschiedenen Rollen innerhalb eines Softwareprojekts wahrzunehmen</li><li>• Fähigkeit die Methoden und Werkzeuge zu bewerten und einzusetzen</li></ul>				

	<ul style="list-style-type: none"> <li>• Einschätzung der eigenen Kompetenz und Leistungsfähigkeit in realitätsnahen Situationen</li> <li>• Training der Soft Skills, insbesondere Teamfähigkeit</li> <li>• Kommunikation mit Kunden</li> <li>• Präsentationsfähigkeit</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: <ul style="list-style-type: none"> <li>• Der vorherige oder parallele Besuch weiterführender Veranstaltungen aus dem Bereich Software Engineering</li> <li>• Empfehlenswert ist Praxiserfahrung</li> </ul>
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0359-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0359-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Text Analytics					
<b>Modul Nr.</b> 20-00-0596	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0596-se	Text Analytics	3	Seminar	2
2	<b>Lerninhalt</b> Die Seminarreihe beschäftigt sich mit aktuellen Themen in der automatischen Sprachverarbeitung. Es werden grundlegende Methoden und Technologien zur Analyse geschriebener, natürlicher Sprache vorgestellt, wobei der Schwerpunkt des Seminars in jedem Semester neu gesetzt wird.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung besucht haben, können sie <ul style="list-style-type: none"><li>• aktuelle Forschungsfragen zum Seminarthema benennen und erläutern,</li><li>• wissenschaftliche Veröffentlichungen verstehen, kritisch beurteilen und untereinander diskutieren,</li><li>• ein Forschungsthema eigenständig aufarbeiten und</li><li>• dieses der Gruppe vorstellen und auf Rückfragen und Diskussionsbeiträge eingehen.</li></ul>				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0596-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0596-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Wird in der Veranstaltung bekanntgegeben.
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Fortgeschrittene Themen in Computer Vision und Maschinellem Lernen					
<b>Modul Nr.</b> 20-00-0645	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0645-se	Fortgeschrittene Themen in Computer Vision und Maschinellem Lernen	3	Seminar	2
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Grundlagen der wissenschaftlichen Vortragstechnik und Begutachtung</li><li>• Eigenständiges Einarbeiten in aktuelle Publikationen in Computer Vision oder Maschinellem Lernen (englischsprachig)</li><li>• Eigene darüber hinausgehende Recherche zur Hintergrund-Literatur, angeleitet von Betreuer</li><li>• Erstellen eines zweiteiligen Vortrags (Problemstellung und Lösungsansatz) über eine Publikationen einschließlich Folienpräsentation, angeleitet durch Betreuer</li><li>• Erstellen eines (simulierten) wissenschaftlichen Gutachtens über eine zweite Publikation, angeleitet durch Betreuer</li><li>• Halten des Vortrags vor einem Publikum mit heterogenem Vorwissen</li><li>• Führung der Fachdiskussion nach beiden Vortragsteilen</li><li>• Aktive Teilnahme an den Fachdiskussionen, sowie Feedback an die Vortragenden</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden sich eigenständig in aktuelle Themen der Computer Vision und/oder des Maschinellen Lernens anhand von wissenschaftlichen Veröffentlichungen einarbeiten. Sie können die wesentlichen Beiträge der untersuchten Publikationen erkennen und diese kompakt einem Publikum mit heterogenem Vorwissensstand präsentieren, unter Berücksichtigung von Prinzipien des guten wissenschaftlichen Vortrags. Nach dem Vortrag können die Vortragenden aktiv eine Fachdiskussion zu dem von ihnen präsentierten Thema bestreiten. Weiterhin sind sie in der Lage ein wissenschaftliches Gutachten über eine aktuelle Publikation anzufertigen, welches den üblichen Standards des wissenschaftlichen Begutachtungsprozesses genügt.</p>				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Teilnehmer sollten Grundkenntnisse in Computer Vision, sowie idealerweise maschinellem Lernen besitzen (z.B. durch Besuch von „Computer Vision I“ und „Statistisches Maschinelles Lernen“).
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0645-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0645-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Aktuelle Publikationen, überwiegend des vergangenen Jahres
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Symbolische Ausführung					
Modul Nr. 20-00-0702	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0702-se	Symbolische Ausführung	3	Seminar	2
2	<b>Lerninhalt</b> Symbolische Ausführung von Programmen ist eine fundamentale Analysetechnik, die u.a. die Basis von Testgenerierung, Compileroptimierung, Verifikation oder Visualisierung darstellt. In den letzten Jahren wurden darin bedeutende Fortschritte erzielt. Im Seminar werden die wichtigsten klassischen und neuen Arbeiten zur symbolischen Ausführung vorgestellt.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach Besuch der Veranstaltung verstehen die Teilnehmenden, was die Möglichkeiten und Grenzen dieser fundamentalen Programmanalysetechnik sind.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0702-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0702-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik				



	M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Automatische Code Generierung					
<b>Modul Nr.</b> 20-00-0790	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0790-se	Automatische Code Generierung	4	Seminar	2
<b>2</b>	<b>Lerninhalt</b> - Beispiele von Domänen spezifischen Sprachen - Automatisches Differenzieren - Automatische Erzeugung für an bestimmte Hardwarearchitekturen angepassten Code				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> - Grundkenntnisse über die Entwicklung und Verwendung von DSL. - Kennenlernen einiger Frameworks für DSL und zur Unterstützung des Model Driven Software Developments (MDSD). - Grundlagen des automatischen Differenzierens und dessen Implementierung				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundkenntnisse der Differentialalgebra im Hinblick auf Ableitungen (Kettenregel, Gradienten, Jacobi-Matrix)				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0790-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>[20-00-0790-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Typsysteme von Programmiersprachen					
Modul Nr. 20-00-0796	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0796-se	Typsysteme von Programmiersprachen	3	Seminar	2
2	<b>Lerninhalt</b> Typsysteme sind formale Methoden, die sicherstellen, dass sich Programme gemäß einer Spezifikation korrekt verhalten. Ihr Anwendungsgebiet ist breit gefächert und umfasst bspw. Softwareentwicklung, Programmiersprachendesign und IT Sicherheit. Dieses Seminar beschäftigt sich mit grundlegenden und aktuellen Forschungsthemen zu Typsystemen, zum Beispiel Dependent Types, Typinferenz, Verfahren zur Typprüfung, constraint-basierte Ansätze usw.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Neben praktischen Erfahrungen im wissenschaftlichen Arbeiten erlangen die Studierenden ein tieferes Verständnis für aktuelle Entwicklungen und Herausforderungen in der Forschung an Typsystemen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundkenntnisse in Mathematik und formalen Methoden der Informatik				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0796-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-0796-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Parallel Computing					
Modul Nr. 20-00-0994	Leistungspunkte 4 CP	Arbeitsaufwand 120 h	Selbststudium 75 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0994-se	Parallel Computing	4	Seminar	3
2	Lerninhalt Aktuelle Trends in der Parallelverarbeitung, z.B. <ul style="list-style-type: none"><li>• Neue Anwendungsfelder (z.B. Deep-Learning)</li><li>• Neue parallele Programmiermodelle</li><li>• Entwicklung paralleler Software für Smartphones</li><li>• GPUs, Manycore-Architecturen</li><li>• FPGAs</li><li>• Architekturen für die Post-Moore-Ära</li><li>• Parallele Dateisysteme</li><li>• Neue parallele Algorithmen</li><li>• Exascale-Computing</li><li>• Cloud-Computing</li></ul>				
3	Qualifikationsziele / Lernergebnisse <ul style="list-style-type: none"><li>• Kennenlernen aktueller Themen im Bereich Parallelverarbeitung</li><li>• Literatur auswählen und analysieren</li><li>• Verständliche Berichte formulieren</li><li>• Übersichtliche Folien erstellen</li><li>• Mündlich präsentieren</li><li>• Anderen Teilnehmern Feedback geben</li></ul>				
4	Voraussetzung für die Teilnahme Empfohlen: <ul style="list-style-type: none"><li>• Kennnisse in Rechnerarchitektur, Programmierung, Softwaretechnik</li><li>• Grundlagen paralleler Systeme</li></ul>				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0994-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0994-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Grundlagen statischer Analysen					
<b>Modul Nr.</b> 20-00-1028	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1028-se	Grundlagen statischer Analysen	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Die Grundlagen statischer Analysen, die zur Implementierung von fortgeschrittenen Qualitäts- und Sicherheitsanalysen gebraucht werden.  Exemplarische Auswahl der Themen: - Berechnung von Kontrol- und Datenabhängigkeiten in der Gegenwart von unendlichen Schleifen und nicht reduzierbarer Kontrollflussgraphen. - Slicing von Code - Identifikation von Schleifen in Maschinencode - Konstruktion von Aufrufgraphen - Statische Analyse Frameworks (z.B., IDE, IFDS, Reactive Async) - "Self-Adaptation" und statische Analysen - Sound(iness) - Specification Mining				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Studierenden werden vertraut sein mit den Grundlagen von fortgeschrittenen Analysen und werden in der Lage sein, die Angemessenheit bestimmter Techniken und Algorithmen für konkrete Anwendungsfälle zu beurteilen. Die Studierenden werden weiterhin in der Lage sein fortgeschrittene, technische Themen im Bereich statische Analyse effektiv zu präsentieren.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Das Seminar richtet sich an fortgeschrittene Bachelor- und Masterstudierende. Vertrautheit mit den Grundlagen des Compilerbaus (z.B. SSA Form) ist sehr empfehlenswert.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1028-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul>				



	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1028-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Performance Engineering					
Modul Nr. 20-00-1038	Leistungspunkte 4 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1038-se	Performance Engineering	4	Seminar	2
2	Lerninhalt - Architektur & Eigenschaften von shared-memory multiprocessor(SMP) Maschinen - Messbarkeit & Verständnis von Performanz auf SMP Maschinen - Erste Erfahrung in der Benutzung von ausgewählten Performance Analyse Tools				
3	Qualifikationsziele / Lernergebnisse - Verstaendnis der Performanzfaktoren und -indikatoren von SMP Plattformen: Compute Units, Memory Design, Synchronisation Protocols - Verständnis der Wichtigkeit vergleichbarer, nachvollziehbarer und reproduzierbarer Messergebnisse				
4	Voraussetzung für die Teilnahme Empfohlen:  Grundlagen von C++ und OpenMP				
5	Prüfungsform Bausteinbegleitende Prüfung:  • [20-00-1038-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%)				
7	Benotung Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>[20-00-1038-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Algorithmen und Plattformen des Reinforcement Learning					
Modul Nr. 20-00-1050	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1050-se	Algorithmen und Plattformen des Reinforcement Learning	3	Seminar	2
2	<b>Lerninhalt</b> Im Rahmen dieses Seminars werden wir Reinforcement Learning Algorithmen und deren Anwendung in Intelligenten Technischen Systemen diskutieren. Hierbei sollen Studenten die Fähigkeit erwerben, sich einen unbekannten Text selbstständig zu erarbeiten, für eine Präsentation aufzubereiten und einem Fachpublikum zu präsentieren.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Am Ende dieses Kurses verstehen Studierende die aktuellen Forschungsthemen im Reinforcement Learning und sind in der Lage die Literaturvorstudie für eine Forschungsarbeit in diesem Bereich durchzuführen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Gleichzeitige oder vorhergehende Belegung der Vorlesung "Reinforcement Learning: Von Grundlagen zu den Tiefen Ansätzen" oder "Lernende Roboter" oder vergleichbarer Veranstaltungen.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1050-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b>				

	Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1050-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Erweitertes Seminar - Systems and Machine Learning					
<b>Modul Nr.</b> 20-00-1057	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 120 h	<b>Selbststudium</b> 75 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1057-se	Erweitertes Seminar - Systems and Machine Learning	4	Seminar	3
<b>2</b>	<b>Lerninhalt</b> <p>Dieses Seminar dient der Diskussion neuer Forschungspapiere im Zusammenhang von Hardware-/Softwaresystemen und maschinellem Lernen (ML). Das Seminar zielt auf die Verbindungen zwischen diesen Themenbereichen ab und diskutiert Fragestellungen, die auf praktisch anwendbares maschinelles Lernen zugeschnitten sind wie z.B. Hardware-Beschleuniger für ML, verteilte skalierbare ML-Systeme, neuer Programmierparadigmen für ML, Automatisiertes ML, sowie Anwendungen von ML für Systeme.</p> <p>Jeder Teilnehmer/jede Teilnehmerin präsentiert ein Forschungspapier, das anschließend von allen Teilnehmenden diskutiert wird. Darüber hinaus werden zusammenfassende Arbeiten in Gruppen verfasst und einem Peer-Review Prozess unterzogen. Die vorzustellenden Arbeiten stellen in der Regel aktuelle Publikationen in relevanten Konferenzen und Zeitschriften dar.</p> <p>Das Seminar wird als Blockveranstaltung angeboten.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach diesem Seminar sind Studierende in der Lage:</p> <ul style="list-style-type: none"><li>- einen unbekannten Text aus den Bereichen des Seminars selbständig aufzuarbeiten</li><li>- eine Präsentation und eine schriftliche Zusammenfassung für ein Fachpublikum in diesem Gebiet zu entwickeln</li><li>- an einer Fachdiskussion über ein Thema aus den Bereichen des Seminars sinnvoll teilzunehmen</li><li>- die Meinung über eine wissenschaftliche Arbeit in der Form eines schriftlichen Peer-Reviews zu artikulieren</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Grundkenntnisse in maschinellem Lernen, skalierbarem Datenmanagement und Hardware-/Softwaresystemen.</p>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1057-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1057-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Deep Learning und Digital Humanities					
<b>Modul Nr.</b> 20-00-1080	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1080-se	Deep Learning und Digital Humanities	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Der Fokus des Seminars wird auf Humanities Anwendungen wie Gedicht-Generierung und Analyse, Metaphern- und Emotions-Identifikation, etc. liegen, und wie diese mithilfe von Deep Learning gelöst werden können. Die Studierenden werden Paper lesen und diese während des Seminars präsentieren.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach diesem Seminar werden Studierende dazu in der Lage sein: * Probleme aus dem Umfeld von Digital Humanities zu verstehen * verstehen, wie Deep Learning verwendet werden kann, um diese zu lösen * verstehen, wie man crowd-sourcing für Annotationen durchführt				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Veranstaltungen aus dem Bereich des Deep Learning sind hilfreich, aber nicht vorausgesetzt				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1080-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				



	<ul style="list-style-type: none"> <li>• [20-00-1080-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Software-Engineering für Künstliche Intelligenz					
<b>Modul Nr.</b> 20-00-1097	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1097-se	Software-Engineering für Künstliche Intelligenz	4	Seminar	3
<b>2</b>	<b>Lerninhalt</b> <p>Künstliche Intelligenz (KI) ist mittlerweile Bestandteil vieler datengetriebenen Anwendungen; zum Beispiel in der Finanzindustrie, Medizin, Kognitionswissenschaft oder Biologie. Derartige Ansätze des maschinellen Lernens (ML) erfordern eine genaue Domänen- und Anforderungsanalyse, angemessenes Softwaredesign und -Entwicklung, besonderes Testen und Debugging sowie spezielle Techniken, um Skalierbarkeit und Wartbarkeit sicherzustellen. Während KI-Systeme zunehmend größeren Einfluss in vielen Bereichen besitzen, verwenden Entwickler und Data-Scientists weiterhin Methoden (Scripting, informelle/nicht-verschriftlichte Spezifikationen, trial-and-error Testing), die nicht dem aktuellen Stand der Technik in den Ingenieursdisziplinen entsprechen. Vor diesem Hintergrund ist es von entscheidender Bedeutung die Jahrzehnte lange Entwicklung im Software-Engineering (SE) zur Systematisierung von Entwicklungsprozessen für diesen Bereich zu nutzen.</p> <p>In diesem Kurs wird Studierenden ein Thema im Bereich SE für KI zugewiesen. Ausgehend von vorgegebenen Quellen und persönlicher erweiternder Literaturrecherche bereiten Studierende eine Präsentation mit anschließender Diskussion vor. Diese werden an regelmäßigen Terminen gehalten. Alle Studierenden, die an einem Termin nicht präsentieren, bereiten sich auf die jeweilige Diskussion mit einführendem Lesematerial vor. Die Benotung basiert auf der Vorbereitung und der Präsentation der zugewiesenen Themenschwerpunkte sowie auf der Teilnahme an allen Diskussionen.</p> <p>Beachten Sie bitte die Kursseite für mehr Informationen und Ankündigungen: <a href="https://allprojects.github.io/SE4AI/">https://allprojects.github.io/SE4AI/</a></p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Studierenden entwickeln ein tieferes Verständnis zu SE für KI. Dies umfasst die Schwerpunkte Requirements Engineering, Qualitätssicherung, Entwicklungsprozesse sowie Softwarearchitektur und -Design für Modularität, Wiederverwendbarkeit, Effizienz, Skalierbarkeit, Fairness und Privatsphäre.				

	Die Studierenden lernen die Vorbereitung und Präsentation von wissenschaftlichen Inhalten für ein Publikum mit unterschiedlichem Hintergrundwissen. Außerdem üben die Studierenden die effiziente Vorbereitung von und aktive Teilnahme an wissenschaftlichen Diskussionen sowie deren Moderation.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Basiswissen zu Software-Engineering. Interesse an Künstlicher Intelligenz.
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1097-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1097-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Data Science and Engineering**

**Wahlbereich Studienbegleitende Leistungen**

**Praktikum in der Lehre**

## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Softwaretechnik					
<b>Modul Nr.</b> 20-00-0443	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0443-pl	Praktikum in der Lehre - Softwaretechnik	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Vorbereitung und Korrektur von Übungen, Abhalten von Übungsstunden, Betreuung von Praktischen Übungen.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Es sollen Fähigkeiten erlernt werden, geeignete Lernmaterialien für Schulungen in Informatikthemen selbst zu erstellen, ihren Einsatz kritisch zu begleiten und dabei auch die Lernenden zu betreuen und anzuleiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b>				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0443-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0443-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Data Management					
<b>Modul Nr.</b> 20-00-1040	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1040-pl	Praktikum in der Lehre - Data Management	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Erstellung von Übungs- und Vorlesungsmaterial				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Erfahrung in der Betreuung von Studierenden im Themenbereich Datenmanagement, mit dem Fokus auf das neu erstellte Übungs- und Vorlesungsmaterial				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch der Veranstaltung „Informationsmanagement“ oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1040-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1040-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Deep Learning for Natural Language Processing					
<b>Modul Nr.</b> 20-00-1044	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1044-pl	Praktikum in der Lehre - Deep Learning for Natural Language Processing	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Vorbereitung, Abhalten und Korrektur eines Shared Tasks. Bei einem Shared Task erhalten die Studierenden ein aktuelles Forschungsproblem und müssen für dieses die Methoden aus der Vorlesung nutzen um innovative Lösungen zu entwickeln. Die Lösungen können quantitativ miteinander verglichen werden, um die beste Lösung zu identifizieren. Die Aufgabe ist es einen entsprechenden Datensatz auszuwählen und vorzubereiten, die Studierenden in die Aufgabe einzuführen sowie die abschließende quantitative und qualitative Bewertung der entwickelten Systeme. Während des Shared Tasks müssen Rückfragen beantwortet werden und falls nötig individuelle Hilfe angeboten werden. Neben dem Shared Task erfolgt eine Unterstützung bei den wöchentlichen Übungen, beispielsweise für die Beantwortung von Fragen zu den Hausübungen oder Unterstützung bei der Korrektur von Übungen.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> In einem Praktikum der Lehre bearbeiten die Studierenden Probleme, die sowohl fachliche als auch didaktische Aspekte haben und wirken an der Umsetzung der von ihnen erarbeitete Resultate mit.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Der vorherige Besuche der Veranstaltung “Deep Learning for Natural Language Processing” oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1044-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1044-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum in der Lehre - Statistisches Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-1070	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1070-pl	Praktikum in der Lehre - Statistisches Maschinelles Lernen	5	Praktikum in der Lehre	3
2	<b>Lerninhalt</b> Unterstützung der Lehre wie z.B., Betreuung von Übungsgruppen, Sprechstunden, o.ä.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Vorbereitung auf eigenständige Lehrtätigkeit.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreiche Absolvierung der Veranstaltung „Statistisches Maschinelles Lernen“ oder entsprechende Kenntnisse.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1070-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1070-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				
8	<b>Verwendbarkeit des Moduls</b>				

	B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum in der Lehre - Foundations of Language Technology (FOLT)					
<b>Modul Nr.</b> 20-00-1110	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Data Science and Engineering		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1110-pl	Praktikum in der Lehre - Foundations of Language Technology (FOLT)	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Vorbereitung, Abhalten und Korrektur eines Shared Tasks, Anbieten von Sprechstunden für die Lerninhalte, Halten von Tutorien, und vergleichbare Aufgaben für die Lehre				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Abschluss der Veranstaltung sind die Studierenden in der Lage, eigenständig ein Tutorium zu veranstalten, eine Shared Task vorzubereiten und vergleichbare Aufgaben der Lehre zu übernehmen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen wird das vorherige Belegen von „Foundations of Language Technology“ (FOLT) oder vergleichbarer Kurse (z.B. „Deep Learning for Natural Language Processing“ (DL4NLP))				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1110-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1110-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Distributed Computing**

**Wahlbereich Computer Networks and  
Distributed Systems**

## Modulbeschreibung

<b>Modulname</b> TK1: Verteilte Systeme und Algorithmen					
<b>Modul Nr.</b> 20-00-0065	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0065-iv	TK1: Verteilte Systeme und Algorithmen	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Lernziele:  • Umfassendes Überblickswissen über die grundlegenden Probleme und Ansätze • Tiefgehendes Methodenwissen zu klassischen verteilten Algorithmen und Programmierparadigmen • Anwendbare exemplarische Kenntnis aktueller Entwicklungen und Standards  Stoffplan:  • Einführung • Auffrischung und Ergänzung von Kapitel 1 der Kanonik Net-Centric Computing • Überblick über die Vorlesung • Verteilte Algorithmen  ○ Elementaralgorithmen (z.B. globaler Zustand) ○ Basisalgorithmen (z.B. Ausschluss, Konsens, Kooperation) ○ Formalisierung (Eigenschaften und deren Nachweis)  • Verteiltes Programmieren  ○ Push-Paradigmen (z.B. IPC, RPC, DOC) ○ aktuelle Ansätze (z.B. Pull-Paradigmen, Objektmobilität)				
3	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung die Grundlagen der verteilten Programmierung und verteilter Algorithmen. Sie verstehen die grundlegenden				



	Probleme verteilter Systeme und die klassischen verteilten Algorithmen und Programmierparadigmen. Sie können klassische und aktuelle Standards verteilter Programmierung praktisch anwenden.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: „Computer Netze und verteilte Systeme“
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0065-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0065-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein: <ul style="list-style-type: none"> <li>George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems. Concepts and Design (Gebundene Ausgabe) 832 Seiten, Addison Wesley; Auflage: 4th (14. Juni 2005), ISBN: 0321263545</li> <li>M. Boger: Java in verteilten Systemen, 1999, dpunkt-Verlag, Heidelberg, ISBN: 3932588320</li> </ul>

	<ul style="list-style-type: none"> <li>• G. Tel: Introduction to Distributed Algorithms, 2nd Ed 2001, Cambridge University Press, ISBN: 0521794838</li> <li>• A. Tanenbaum, M.v.Steen, Verteilte Systeme: Grundlagen und Paradigmen, Pearson Studium 2003, ISBN: 3827370574</li> <li>• A. Tanenbaum: Computernetzwerke. 4te Auflage. Pearson Studium 2003, ISBN-10: 3827370469</li> <li>• J. Kurose, K. Ross: Computer Networking, 1. Ed. 2000, Adison-Wesley. ISBN: 0201477114</li> <li>• L. Peterson, B. Davie, Computernetze, 1. Aufl. 2000, dpunkt Heidelberg, ISBN: 393258869X</li> <li>• Hammerschall, U.: Verteilte Systeme und Anwendungen. Pearson, München 2005, ISBN: 3827370965</li> </ul>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Netzsicherheit					
<b>Modul Nr.</b> 20-00-0512	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0512-iv	Netzsicherheit	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Die integrierte Veranstaltung Netzsicherheit umfasst Sicherheits-Prinzipien und -Praxis in Telekommunikationsnetzen und dem Internet. Die grundlegenden Verfahren aus dem Bereich IT Sicherheit und Kryptographie werden auf den Bereich der Kommunikationsnetze übertragen. Hierbei verfolgen wir einen Top-down Ansatz. Beginnend mit der Anwendungsschicht erfolgt eine detaillierte Betrachtung von Prinzipien und Protokollen zur Absicherung von Netzen. Ergänzend zu etablierten Mechanismen werden ausgewählte aktuelle Entwicklungen im Bereich Netzsicherheit erläutert. Lerninhalte: - Netzsicherheit: Einführung, Motivation und Herausforderungen - Grundlagen: Ein Referenzmodell für Netzsicherheit, Sicherheitsstandards für Netze und das Internet, Bedrohungen, Angriffe, Sicherheitsdienste und -mechanismen - Kryptographische Grundlagen zur Absicherung von Netzen: Symmetrische Kryptographie und deren Anwendung in Netzen, asymmetrische Kryptographie und deren Anwendung in Netzen, unterstützende Mechanismen zur Implementierung von Sicherheitslösungen - Sicherheit auf der Anwendungsschicht - Sicherheit auf der Transportschicht - Sicherheit auf der Vermittlungsschicht - Sicherheit auf der Sicherungsschicht - Sicherheit auf der Bitübertragungsschicht und physische Sicherheit - Angewandte Netzsicherheit: Firewalls, Intrusion Detection Systeme - Ausgewählte Themen der Netzsicherheit				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung haben die Studierenden ein umfassendes Wissen auf dem Gebiet der Netzsicherheit mit dem Schwerpunkt auf Internetsicherheit. Sie können die wichtigsten Grundlagen der IT Sicherheit sowie der Kryptographie auf den Bereich Kommunikationsnetze übertragen und anwenden. Die Studierenden können die wichtigsten Basistechnologien zur Absicherung von Netzen				

	<p>unterscheiden. Sie weisen ein tiefgehendes Verständnis von Sicherheitsmechanismen auf den unterschiedlichen Protokollschichten auf (Anwendungsschicht, Transportschicht, Vermittlungsschicht, Sicherungsschicht, physikalische Schicht). Somit sind sie in der Lage, die Charakteristiken und Grundprinzipien des Problemraumes Netzsicherheit detailliert zu erläutern und weisen auf diesem Feld ein fundiertes Wissen in Praxis und Theorie auf. Darüber hinaus können sie aktuelle Entwicklungen im Bereich Netzsicherheit erläutern (z.B. Sicherheit in peer-to-peer Systemen, Sicherheit in mobilen Netzen, etc.). Die Übung vertieft das theoretische Wissen durch Literatur-, Rechen- und praktische Implementierungs-/Anwendungsübungen.</p>
4	<p><b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen der IT-Sicherheit, Kryptographie und Kommunikationsnetze</p>
5	<p><b>Prüfungsform</b> Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0512-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b> Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0512-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>

<b>9</b>	<b>Literatur</b> Charlie Kaufman, Radia Perlman, Mike Speciner: Network Security – Private Communication in a Public World, 2nd Edition, Prentice Hall, 2002, ISBN: 978-0-14-046019-6; weiterhin ausgewählte Buchkapitel und ausgewählte wissenschaftliche Veröffentlichungen
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Sichere Mobile Systeme					
<b>Modul Nr.</b> 20-00-0583	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0583-iv	Sichere Mobile Systeme	3	integrierte Veranstaltung	2
<b>2</b>	<b>Lerninhalt</b> Die integrierte Veranstaltung Sichere Mobile Systeme befasst sich mit Fragen zur Sicherheit in drahtlosen und Mobilien Netzen und Kommunikationssystemen. Grundlagen der Thematik werden durch aktuelle Forschungsthemen ergänzt. Lerninhalte: - Sicherheitsbetrachtung und Modellierung von Bedrohungen bei mobilen und drahtlosen Systemen - Ausgewählte Angriffe und Sicherheitsmechanismen spezifisch für mobile und drahtlosen Systeme - Sicherheit in drahtlosen Sensornetzen - Sicherheit in drahtlosen Mesh-Netzen - Bedrohungen und Schutz der Privatsphäre in mobilen und drahtlosen Systemen - Sicherheit in zellularen Netzen (GSM, UMTS, LTE) - Sicherheit auf der Bitübertragungsschicht - Ausgewählte Forschungsthemen in mobilen und drahtlosen Systemen				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung besitzen die Studierenden ein spezialisiertes Wissen auf dem Gebiet der Sicherheit in mobilen, verteilten, drahtlosen Netzen mit dem Schwerpunkt auf Internetsicherheit. Sie können die wichtigsten Grundlagen der IT Sicherheit, der Kryptographie sowie der Netzsicherheit in klassischen Netzen auf mobile Systeme übertragen und anwenden.  Die Studierenden weisen ein tiefgehendes Verständnis von Sicherheitsmechanismen auf den unterschiedlichen Protokollschichten auf (Anwendungsschicht, Transportschicht, Vermittlungsschicht, Sicherungsschicht, physikalische Schicht). Somit sind sie in der Lage,				

	die Charakteristiken und Grundprinzipien des Problemraumes zu erfassen und weisen auf dem Feld sicherer mobiler Systeme ein fundiertes Wissen in Praxis und Theorie auf.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen der Netzsicherheit und der Mobilen Netze
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0583-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0583-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Levente Buttyan, Jean-Pierre Hubaux: Security and Cooperation in Wireless Networks, Cambridge University Press, 2008, ISBN: 978-0-521-87371-0 (book is available online for download).  Ausgewählte Buchkapitel und ausgewählte wissenschaftliche Veröffentlichungen.
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Mobile Netze					
<b>Modul Nr.</b> 20-00-0748	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0748-iv	Mobile Netze	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b>				
	Mobilkommunikation und drahtlose Kommunikationstechniken haben sich in den letzten Jahren rapide weiterentwickelt. Die integrierte Veranstaltung erläutert Charakteristiken und Grundprinzipien mobiler Netze, und praktische Lösungsansätze werden vorgestellt. Der Fokus der Veranstaltung liegt hierbei auf der Vermittlungsschicht (Netzwerkschicht). Zusätzlich zum Stand der Technik werden in der Veranstaltung aktuelle Forschungsfragen diskutiert und Methoden und Werkzeuge zur systematischen Behandlung dieser Fragen erläutert. Die Inhalte werden in Übungseinheiten vertieft.				
	Lerninhalte:				
	- Einleitung: Drahtlose und mobile Kommunikation: Anwendungen, Geschichte, Marktchancen				
	- Überblick über drahtlose Kommunikation: Drahtlose Übertragung, Frequenzen und Frequenzregulierung, Signale, Antennen, Signalausbreitung, Multiplex, Modulation, Spreizband-Technik, Zellulare Systeme				
	- Medienzugriff: SDMA, FDMA, CDMA, TDMA (Feste Zuordnung, Aloha, CSMA, DAMA, PRMA, MACA, Kollisionsvermeidung, Polling)				
	- Drahtlose Lokale Netze (Wireless LAN): IEEE 802.11 Standard inklusive Bitübertragungsschicht, Sicherungsschicht und Zugriffverfahren, Dienstgüte, Energieverwaltung				
	- Drahtlose Stadtnetze, drahtlose Mesh Netze, IEEE 802.16 Standard inklusive Betriebsmodi, Medienzugriff, Dienstgüte, Ablaufkoordination				
	- Mobilität auf der Netzwerkschicht: Konzepte zur Mobilitätsunterstützung, Mobile IP				
	- Ad hoc Netze: Terminologie, Grundlagen und Applikationen, Charakteristika von Ad hoc Kommunikation, Ad hoc Routing Paradigmen und Protokolle				
	- Leistungsbewertung von mobilen Netzen: Einführung in die Leistungsbewertung, systematischer Ansatz/häufige Fehler und wie man sie vermeiden kann, experimentelles Design und Analyse				
	- Mobilität auf der Transportschicht: Varianten von TCP (Indirect TCP, Snoop TCP, Mobile TCP, Wireless TCP)				



	- Mobilität auf der Anwendungsschicht: Anwendungen für mobile Netze und drahtlose Sensornetze
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung haben Studierende ein umfassendes Wissen der Funktionsweise mobiler Kommunikationsnetze. Sie können die wichtigsten Grundlagen drahtloser Kommunikationstechniken erläutern. Die Studierenden können weiterhin Medienzugriffsverfahren kategorisieren und die Funktionsweise dieser Verfahren im Detail erklären. Insbesondere weisen sie ein tiefgehendes Verständnis von Verfahren auf Vermittlungsschicht und Transportschicht auf, mit Schwerpunktsetzung auf Ad hoc und Mesh Netze. Die Studierenden erlangen Wissen über die Zusammenhänge zwischen unterschiedlichen Protokollschichten und können ihr erworbenes Wissen auf die methodische Analyse von realen Kommunikationssystemen anwenden. Sie sind somit in der Lage, die Charakteristiken und Grundprinzipien des Problemraumes drahtloser und mobiler Kommunikation detailliert zu erläutern und weisen auf diesem Feld ein fundiertes Wissen in Praxis und Theorie auf. Die Übungsteile der integrierten Veranstaltung vertiefen das theoretische Wissen durch Literatur-, Rechen- und praktische Implementierungs-/Anwendungsübungen.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen der Kommunikationsnetze
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0748-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0748-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science

	M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Ausgewählte Buchkapitel und ausgewählte wissenschaftliche Veröffentlichungen
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Concepts and Technologies for Distributed Systems and Big Data Processing					
<b>Modul Nr.</b> 20-00-0951	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0951-iv	Concepts and Technologies for Distributed Systems and Big Data Processing	3	Integrierte Veranstaltung	2
<b>2</b>	<b>Lerninhalt</b> <p>The course provides an overview of recent advances in distributed systems for Big Data processing. The course starts presenting computational models for high throughput batch processing like MapReduce. Next, we will introduce software engineering techniques for distributed systems such as REST and component-based architectures. We will then cover low latency real time stream processing and complex event processing. Finally, we will present advanced topics in distributed data-intensive systems, such as geodistribution and security.</p> <p>The course focuses both on the fundamental concepts as well as on the concrete technologies and applications of the aforementioned techniques to real-world case studies.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>- The students are familiar with basic concepts and technologies on distributed systems and big data and are able to implement basic cloud based/distributed applications.</p> <p>- The students are familiar with the fundamental computational models behind recent advances in distributed systems, such as models for batch processing of massive data amounts, stream processing and complex event processing.</p> <p>- The students are familiar with selected advanced topics on big data, including security and geolocalization.</p> <p>- The students know about real-world case studies that apply the concepts and the technologies presented during the course.</p>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0951-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0951-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Distributed Computing**

**Wahlbereich Data-Intensive Systems and  
heterogeneous Hardware**

## Modulbeschreibung

<b>Modulname</b> Algorithmen für Hardware-Entwurfswerkzeuge					
<b>Modul Nr.</b> 20-00-0183	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0183-v1	Algorithmen für Hardware-Entwurfswerkzeuge	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> - Das VLSI-Entwurfsproblem - Grundlegende Graphenrepräsentationen und -algorithmen - Darstellung von hierarchischen Schaltungen - Realisierungstechnologien für integrierte Schaltungen - Layout-Kompaktierung - Timing-Analyse - Heuristische Optimierungsverfahren - Platzierungsprobleme, -verfahren und -kostenfunktionen - Exakte Optimierungsverfahren - Partitionierung mit Anwendung in der Platzierung - Floorplanningprobleme, -repräsentationen und -verfahren - Verdrahtungsprobleme, -verfahren und -kostenfunktionen				
3	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung verschiedene Technologien für die Realisierung von integrierten Schaltungen. Sie können aus den verschiedenen Technologien die Anforderungen an Automatisierungswerkzeuge für verschiedene Teilaufgaben des Entwurfs- und Realisierungsprozesses herleiten. Sie sind vertraut mit der Modellierung technologischer Probleme durch formale Konzepte wie Graphen, Gleichungssysteme etc. Sie verstehen grundlegende Verfahren zur Lösung auch von harten Problemen und können aufbauend auf Erfahrungen mit verschiedenen Basisalgorithmen neue bzw. verfeinerte Implementierungen zur Erledigung der Entwurfsaufgaben entwickeln.				
4	<b>Voraussetzung für die Teilnahme</b>				

	Empfohlen: Empfohlen wird der erfolgreiche Besuch der Veranstaltungen “Digitaltechnik” sowie “Algorithmen und Datenstrukturen” und “Funktionale und objektorientierte Programmierung” oder vergleichbarer Veranstaltungen.
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0183-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0183-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein: Gerez: Algorithms for VLSI Design Automation Wang/Chang/Cheng: Electronic Design Automation
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Programmierung Massiv-Paralleler Prozessoren					
<b>Modul Nr.</b> 20-00-0419	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0419-iv	Programmierung Massiv-Paralleler Prozessoren	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> - Grundlagen massiv-paralleler Hardware mit einem Schwerpunkt auf modernen Beschleunigern - parallele Algorithmen - effiziente Programmierung massiv-paralleler Systeme - praktische Programmierprojekte mit Co-Betreuung durch einen Wissenschaftler aus seiner Anwendungsdomain				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung sind Studierende dazu in der Lage, Problemstellungen im Kontext massiv-paralleler Systeme zu analysieren. Sie können selbständig neue Anwendungen entwickeln und ihre Performanz systematisch verbessern. Sie verstehen grundlegende parallele Algorithmen und Programmierparadigmen und können sich selbständig aktuelle Literatur erarbeiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  solide Programmierkenntnisse in C/C++  Kenntnisse in paralleler Programmierung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0419-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				



	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0419-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Fortgeschrittener Compilerbau					
<b>Modul Nr.</b> 20-00-0701	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0701-v1	Fortgeschrittener Compilerbau	5	integrierte Veranstaltung	3
<b>2</b>	<b>Lerninhalt</b> - Compilierung und Laufzeitumgebung für objektorientierte Programmiersprachen - Kontrollflussgraphen als Zwischendarstellung - Statische Datenflußanalyse - Static Single Assignment Form - Eliminierung totaler und partieller Redundanz - Skalare Optimierung - Registerallokation - Ablaufplanung - Schleifenoptimierung - Aufbau realer Compiler (z.B. Phasen, Zwischendarstellung, Compilefluß)				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende verstehen nach erfolgreichem Besuch Techniken für die Übersetzung und Ausführung von objektorientierten Programmen auf Maschinenebene. Sie können die statische Datenflussanalyse auf Kontrollflussgraphen anwenden und sind geübt im praktischen Umgang mit deren SSA-Darstellung. Sie beherrschen Optimierungsverfahren für eine Reihe von Aufgaben sowie fundamentale Verfahren für die Registerallokation. Sie kennen die interne Struktur von realen Compilern für den Produktivbetrieb.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreicher Besuch der Veranstaltung “Einführung in den Compilerbau” oder vergleichbarer Veranstaltungen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0701-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0701-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein:</p> <p>Cooper/Torczon: Engineering a Compiler Muchnick: Advanced Compiler Design and Implementation Aho/Lam/Sethi/Ullman: Compilers - Principles, Techniques, and Tools</p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Advanced Data Management Systems					
Modul Nr. 20-00-1039	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1039-iv	Advanced Data Management Systems	6	Integrierte Veranstaltung	4
2	<b>Lerninhalt</b>  Dies ist eine fortgeschrittene Veranstaltung aus dem Bereich der Architektur und Implementierung moderner Datenbanksysteme mit dem speziellen Fokus auf System-orientieren Aspekten und Interna solcher Systeme. Mögliche Themengebiete die in der Vorlesung behandelt werden sind: moderne Hardwaretechnologien für das Datenbanksysteme, Optimierungen für Hauptspeicherdatenbanken, Parallelisierungsstrategien und Approximative Anfrageausführung usw.  Es wird erwartet, dass für jede Vorlesung aktuelle Veröffentlichungen (SIGMOD, VLDB, etc.) vorher gelesen werden. Die Hauptideen ausgewählter Veröffentlichungen werden in Programmierprojekten umgesetzt.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Abschluss der Veranstaltung haben Studierende ein vertieftes Verständnis von aktuellen Techniken für das Design von modernen Datenbanksystemen. Die Studierenden können Vor- und Nachteile dieser Techniken mit dem Fokus auf möglichen Verbesserungen diskutieren. Sie können einzelne Techniken implementieren und experimentelle Evaluierungen dieser Techniken zum Vergleich von Designalternativen durchführen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Solide Programmierkenntnisse in C and C++ Der vorherige Besuch von „Skalierbare Datenmanagement-Systeme“ und „Informationsmanagement“ oder vergleichbaren Veranstaltungen				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1039-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1039-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Distributed Computing**

**Wahlbereich System Modelling and  
Engineering**

## Modulbeschreibung

Modulname					
Applied Static Analysis					
Modul Nr. 20-00-0949	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0949-iv	Applied Static Analysis	3	Integrierte Veranstaltung	2
2	Lerninhalt				
	Foundations of (scalable) static analyses for large(r) software systems; in particular - Basic Terminology: - AST, SSA, - Object-/ Field-/ Context-/ Flow-/ Path Sensitivity - (I)CFG - Inter-procedural analyses - ... - stack based intermediate representations (JVM Bytecode) - register based intermediate representations (LLVM IR) - program transformations and native code analyses using LLVM Concrete static analyses and algorithms: - Call graph algorithms for libraries and applications - Inter procedural data- and control-flow analyses - IDE/IFDS - Points-to analyses - Escape analyses Applications - General software quality analyses - Capability Analysis - Security Vulnerabilities Detection - Dead Paths/Computations - Next generation software development tools				
3	Qualifikationsziele / Lernergebnisse Students can effectively use the basic static analyses related terminology. Students are familiar with modern static analyses working on intermediate representations. They are able to apply and adapt available static analysis algorithms to new scenarios.				
4	Voraussetzung für die Teilnahme Empfohlen:				

	The lecture is targeted towards Master students with a very high degree of interest in reading, analyzing and also writing code. Basic knowledge in compiler construction is helpful. Deep knowledge of object-oriented programming concepts and in particular of object-oriented programming in Java is required. Interest in learning new programming languages (in particular Scala) is required.
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0949-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0949-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b>					
Fortgeschrittenes Multithreading in C++					
<b>Modul Nr.</b> 20-00-0977	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0977-iv	Fortgeschrittenes Multithreading in C++	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> C++ bietet eine der modernsten Threadschnittstellen, die heute verfügbar sind. Am Beispiel C++ führt dieser Kurs in die fortgeschrittene parallele Programmierung für gemeinsamen Speicher mit Threads ein.  Aufbauend auf den Inhalten der Vorlesung Multithreading in C++ werden die folgenden Themen behandelt: <ul style="list-style-type: none"><li>• C++ Speichermodell und atomare Operationen</li><li>• Entwurf lockfreier nebenläufiger Datenstrukturen</li><li>• Fortgeschrittenes Thread-Management (z.B. Thread Pools)</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung besucht haben, haben Sie erweiterte Kompetenz in der Entwicklung paralleler Programme und sind in der Lage <ul style="list-style-type: none"><li>- Systematisch korrekte und effiziente parallele Programme zu entwickeln</li><li>- Parallele Datenstrukturen zu entwerfen und umzusetzen</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: <ul style="list-style-type: none"><li>• Kenntnisse in C/C++</li><li>• Basiskenntnisse der Programmierung von Threads in C++ (lockbasierte Synchronisation und lockbasierte nebenläufige Datenstrukturen)</li></ul>				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0977-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0977-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Model Checking					
Modul Nr. 20-00-1115	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1115-vl	Model Checking	3	Vorlesung	2
2	Lerninhalt * Temporallogiken: - Lineare temporal Logik (LTL), Computation Tree Logic (CTL) und CTL*: Syntax, Semantik, Komplexität * Modelprüfungsverfahren für LTL, CTL, CTL*, insbesondere Büchiautomaten * Partial Order Reduction * Timed Automata				
3	Qualifikationsziele / Lernergebnisse Im Rahmen des Kurses sollen die Studierenden folgende Fähigkeiten erwerben: * Verständnis der theoretischen Grundlagen der Temporallogiken LTL, CTL und CTL* * Fähigkeit zur Auswahl der geeigneten Logik zur Spezifikation und Modellprüfung in Abhängigkeit von dem zu modellierenden System und der zu prüfenden Eigenschaft * Verfahren und Techniken zur Modellprüfung (Model Checking) z.B. Modellprüfung mit Büchiautomaten, Partial Order Reduction u.ä. * Wissen über die Charakteristika und Grenzen der Modellprüfung * Kenntnisse in der Modellprüfung von Timed Automate * Fähigkeit zur Anwendung von Tools zur Modellprüfung				
4	Voraussetzung für die Teilnahme Empfohlen werden Kenntnisse in * Aussagenlogik * Deduktionssystemen * Automatentheorie				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1115-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%).</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1115-vl] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Konzepte der Programmiersprachen					
Modul Nr. 20-00-1117	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1117-iv	Konzepte der Programmiersprachen	6	Integrierte Veranstaltung	4
2	Lerninhalt Kurze Einführung und Geschichte der Programmiersprachen, Kriterien zur Messung von Programmiersprachen, Grundkonzepte der PL wie Syntax, Semantik, Variablen, Namen, Bindungen, Umfang, Subprogram, Expressionen, Arrays, Pointers, abstrakte Typen, funktionale Programme				
3	Qualifikationsziele / Lernergebnisse Die Studierenden werden am Ende des Kurses in der Lage sein, die zugrundeliegenden Mechanismen der wichtigsten Konzepte hinter Programmiersprachen zu verstehen. Die Studierenden werden auch Erfahrung erhalten, eine einfache Programmiersprache mit einer beliebigen Sprache Workbench namens MPS als Gruppenprojekt zu bauen.				
4	Voraussetzung für die Teilnahme Keine				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1117-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%).				
7	Benotung Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-1117-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Distributed Computing**

**Wahlbereich Studienbegleitende Leistungen**

**Praktika, Projektpraktika und ähnliche  
Veranstaltungen**

## Modulbeschreibung

<b>Modulname</b> Praktikum Algorithmen					
<b>Modul Nr.</b> 20-00-0189	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0189-pr	Praktikum Algorithmen	6	Praktikum	4
2	<b>Lerninhalt</b> Lösung eines algorithmischen Problems aus der Praxis und Umsetzung der Lösung in Software. Konkrete Themenstellung nach Absprache in der Vorbesprechung.				
3	<b>Qualifikationsziele / Lernergebnisse</b> In dieser Veranstaltung erwerben Studierende die Kompetenz zur Lösung algorithmischer Problemstellungen aus der Praxis und die Fähigkeit, Algorithmen in praktisch effiziente Implementationen umzusetzen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  - Kenntnis einer geeigneten Programmiersprache (z.B. Java / C++)  - Vorwissen über grundlegende Algorithmen und Datenstrukturen				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0189-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				



<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0189-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> wird in der Veranstaltung bekannt gegeben
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Implementierung von Programmiersprachen					
<b>Modul Nr.</b> 20-00-0306	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0306-pr	Implementierung von Programmiersprachen	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Es werden Konzepte der Implementierung von Programmiersprachen vermittelt. Ferner werden diese Konzepte angewendet, um Erweiterungen für Programmiersprachen zu implementieren.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Fähigkeit, eine professionelle Aufgabe aus der Informatik selbstständig und erfolgreich nach den anerkannten Grundsätzen der Profession zu bearbeiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Es wird kein Vorwissen vorausgesetzt. Jedoch sind gute Programmiererfahrungen sowie Kenntnisse über Kompilerverbau und virtuelle Maschinen von Vorteil.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0306-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:  • [20-00-0306-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname Praktikum Sichere Mobile Netze					
Modul Nr. 20-00-0552	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0552-pr	Praktikum Sichere Mobile Netze	6	Praktikum	4
2	<b>Lerninhalt</b> Das Praktikum Sichere Mobile Netze behandelt die angewandte Softwareentwicklung und Hardware-Software Entwicklung in den Themenbereichen Kommunikationsnetze, Sicherheit, Mobile Netze und Drahtloser Kommunikation bzw. der Kombination dieser Bereiche. Ziel ist das Lösen einer Problemstellung im Team aus den genannten Bereichen durch Implementierung in Software bzw. Hardware/Software.  Lerninhalte: - Lösen einer Fragestellung im Bereich Kommunikationsnetze, Sicherheit, Mobile Netze und Drahtloser Kommunikation - Recherche von Lösungsalternativen und Abwägung von Vor-/Nachteilen der Alternativen - Konzipieren einer Softwarearchitektur bzw. kombinierten Hardware-Software Architektur - Entwerfen eines auf die Zielplattform angepassten Hardware-/Softwaredesigns - Prototypische Umsetzung auf der ausgewählten Zielplattform - Evaluation des Gesamtsystems in Bezug auf verschiedene Gütemaße - Dokumentation der erstellten Lösung				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung besitzen die Studierenden die Fähigkeit Problemstellungen im Bereich Sichere Mobile Netze softwaretechnisch zu lösen. Die Studierenden haben hierzu Kenntnisse im Entwurf/der Umsetzung komplexer Protokolle bzw. Anwendungen in einem/mehreren der Bereiche Kommunikationsnetze, Sicherheit, Mobile Netze und Drahtloser Kommunikation erlangt. Die Studierenden sind in der Lage die gewählten Protokolle und Anwendungen zu implementieren, zu testen und deren Funktionsfähigkeit und Leistungsfähigkeit zu evaluieren. Sie sind in der Lage die erstellten Softwareartefakte verständlich zu dokumentieren und die erzielten Projektfortschritten und -ergebnissen verständlich zu präsentieren.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Erfolgreiche Teilnahme an einer Integrierten Veranstaltung des Fachgebiets SEEMOO				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0552-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)</p>
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0552-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Themenspezifisch ausgewählte, aktuelle wissenschaftliche Veröffentlichungen
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Projektpraktikum Sichere Mobile Netze					
<b>Modul Nr.</b> 20-00-0553	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0553-PP	Projektpraktikum Sichere Mobile Netze	9	Projektpraktikum	6
<b>2</b>	<b>Lerninhalt</b> Das Projektpraktikum Sichere Mobile Netze behandelt die angewandte Softwareentwicklung und Hardware-Software Entwicklung in den Themenbereichen Kommunikationsnetze, Sicherheit, Mobile Netze und Drahtloser Kommunikation bzw. der Kombination dieser Bereiche. Ziel ist das eigenständige Bearbeiten eines Entwicklungsprojektes im Team.  Lerninhalte: - Eigenständiges Bearbeiten eines Entwicklungsprojektes im Bereich Kommunikationsnetze, Sicherheit, Mobile Netze und Drahtloser Kommunikation - Projektplanung und Projektmanagement - Recherche von Lösungsalternativen und Abwägung von Vor-/Nachteilen der Alternativen - Konzipieren einer Softwarearchitektur bzw. kombinierten Hardware-Software Architektur - Entwerfen eines auf die Zielplattform angepassten Hardware-/Softwaredesigns - Prototypische Umsetzung auf der ausgewählten Zielplattform - Evaluation des Gesamtsystems in Bezug auf verschiedene Gütemaße - Dokumentation der erstellten Lösung sowie ausführliche Dokumentation des Projektmanagements				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung besitzen die Studierenden die Fähigkeit komplexe Problemstellungen im Bereich Sichere Mobile Netze softwaretechnisch zu lösen. Die Studierenden können hierzu eigenständig ein Projekt definieren, verwalten und durchführen. Die Studierenden haben Kenntnisse im Entwurf/der Umsetzung komplexer Protokolle bzw. Anwendungen in einem/mehreren der Bereiche Kommunikationsnetze, Sicherheit, Mobile Netze und Drahtloser Kommunikation erlangt. Die Studierenden sind in der Lage die gewählten Protokolle und Anwendungen zu implementieren, zu testen und deren Funktionsfähigkeit und Leistungsfähigkeit zu evaluieren. Sie sind in der Lage die				

	Projektplanung und -verwaltung sowie die erstellten Softwareartefakte verständlich zu dokumentieren und die erzielten Projektfortschritten und -ergebnissen verständlich zu präsentieren.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreiche Teilnahme an einer Integrierten Veranstaltung des Fachgebiets SEEMOO
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0553-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0553-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Themenspezifisch ausgewählte, aktuelle wissenschaftliche Veröffentlichungen
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum zu Algorithmen für Hardware-Entwurfswerkzeuge					
<b>Modul Nr.</b> 20-00-0571	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0571-pr	Praktikum zu Algorithmen für Hardware-Entwurfswerkzeuge	6	Praktikum	4
2	<b>Lerninhalt</b> - Realisieren von Hardware-Entwurfswerkzeugen aus dem Bereich Layout-Synthese, speziell zu Themen wie Timing Analyse, Platzierung und Verdrahtung - Evaluieren der Ergebnisqualität und Rechenzeit- und Speicheranforderungen der eigenen Werkzeuge im Vergleich zu existierenden Implementierungen				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Absolvieren der Veranstaltung können die Studierenden eigenständig Hardware-Entwurfswerkzeuge für eine vorgegebene Zieltechnologie von integrierten Schaltungen erstellen. Sie können ihre Werkzeuge bezüglich verschiedener Gütemaße evaluieren und mit anderen existierenden Implementierungen vergleichen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der erfolgreiche Besuch bzw. die aktive parallele Teilnahme an der Veranstaltung “Algorithmen für Hardware-Entwurfswerkzeuge” ist dringend empfohlen.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0571-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				



	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0571-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Bereitgestellte wissenschaftliche Arbeiten zu den vorgeschlagenen Basisverfahren.
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Praktikum Compilerbau					
Modul Nr. 20-00-0911	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0911-pr	Praktikum Compilerbau	6	Praktikum	4
2	Lerninhalt Eigenständiges Implementieren eines Compilers bzw. von wesentlichen Teilen davon (z.B. einzelne Optimierungspasses oder Back-Ends).				
3	Qualifikationsziele / Lernergebnisse Nach erfolgreichem Absolvieren der Veranstaltung können die Studierenden wesentliche Teile von modernen Compilern selbständig implementieren und ggf. in existierende Compiler-Frameworks integrieren. Dabei können sie ihre Kenntnisse sowohl von compiler-spezifischem Wissen (beispielsweise über verschiedene Zwischendarstellungen) als auch allgemeinen Programmiertechnik (z.B. Design Patterns) anwenden und vertiefen.				
4	Voraussetzung für die Teilnahme Empfohlen: Erfolgreicher Besuch der Vorlesungen „Rechnerorganisation“, „Einführung in den Compilerbau“ und „Fortgeschrittener Compilerbau“ bzw. entsprechende Kenntnisse aus anderen Veranstaltungen				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0911-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%)				
7	Benotung				

	Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0911-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird jeweils passend für die aktuelle Aufgabenstellung bekanntgegeben (z.B. wissenschaftliche Arbeiten zu Optimierungsverfahren, Beschreibung eines Zielprozessors)
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Embedded System Hands-On 1: Entwurf und Realisierung von Hardware/Software-Systemen					
<b>Modul Nr.</b> 20-00-0959	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0959-pr	Embedded System Hands-On 1: Entwurf und Realisierung von Hardware/Software-Systemen	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Diese Veranstaltung richtet sich an Studierende, die grundlegende praktische Kenntnisse im Entwurf und der Realisierung eingebetteter Systeme erwerben möchten.  Nach der Einführung von wichtigen Konzepten und Techniken wie  - Grundlagen der Elektrotechnik - Umgang mit Laborelektronik - Entwurf und Realisierung von elektronischen Schaltungen - Sensordaten: Erfassung und Verarbeitung - Bus-Systeme in eingebetteten Systemen - Programmieren und Debuggen von heterogenen eingebetteten Systemen - Linux Kernel in eingebetteten Systemen  entwickeln die Teilnehmerinnen und Teilnehmer auf Basis des zuvor Gelernten ein eigenes eingebettetes System.  Dabei stehen verschiedene Projekte zur Auswahl, welche je nach eigenen Interessen eine Fokussierung auf die Software- oder die Hardware-Entwicklung erlauben.				
	<b>3</b> <b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme sind Studierende mit dem praktischen Entwurf und der Realisierung von eingebetteten Hardware/Software-Systemen vertraut.  Dazu gehören auch Kenntnisse von elektrotechnischen Grundlagen und der Umgang mit Laborelektronik, die Verwendung von Beschreibungssprachen und EDA/CAD-Werkzeugen für den Hardware-Entwurf, das Programmieren und Debuggen speziell im Umfeld eingebetteter Systeme sowie auch der Einsatz von Linux als Betriebssystem in diesem Kontext.				

<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreicher Besuch der Vorlesungen „Digitaltechnik“, „Rechnerorganisation“, „Architektur und Entwurf von Rechnersystemen“, „Betriebssysteme“ und „Parallele Programmierung“ oder vergleichbare Kenntnisse aus anderen Veranstaltungen.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0959-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0959-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Embedded Systems Hands-On 2: Entwurf von Hardware-Beschleunigern für Systems-on-Chip					
<b>Modul Nr.</b> 20-00-0968	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0968-pr	Embedded Systems Hands-On 2: Entwurf von Hardware-Beschleunigern für Systems-on-Chip	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> <p>Diese Veranstaltung richtet sich an Studierende, die grundlegende Kenntnisse im Design von Hardwarebeschleunigern im Rahmen eines Systems-on-Chip erhalten möchten.</p> <p>Im Rahmen des Praktikums erhalten Studierende umfangreiche Einblicke in relevante Themen wie:</p> <ul style="list-style-type: none"><li>- Treiber für selbst erstellte Hardwarebeschleuniger</li><li>- Einbindung von in Bluespec erstellten Beschleunigern in ein Zynq SoC</li><li>- Toolchains für Hardware- und Software-Komponenten</li></ul> <p>Die Teilnehmer werden im Rahmen des Praktikums Aufgaben zu einem typischen Einsatzgebiet von Hardwarebeschleunigung bearbeiten. Ein typisches Anwendungsgebiet eines solchen Hardwarebeschleunigers ist die Verarbeitung und Erfassung von Kamerabildern, zum Beispiel im Rahmen von Stereo Vision.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Die Teilnehmenden erwerben die Fertigkeiten, das in vorangehenden Veranstaltungen erworbene Methodenwissen nun anzuwenden, um ein eingebettetes System mittels Hardware/Software-Co-Entwurf zu realisieren.</p>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen:</p> <p>Grundlegende Kenntnisse im Umgang mit embedded Linux zum Beispiel aus „Embedded Systems Hands-On 1“</p> <p>Bluespec SystemVerilog aus „Architektur und Entwurf von Rechnersystemen“</p>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0968-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0968-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Distributed Systems Programming: Projektpraktikum					
<b>Modul Nr.</b> 20-00-0984	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0984-pp	Distributed Systems Programming: Projektpraktikum	9	Praktikum	6
<b>2</b>	<b>Lerninhalt</b> Das "DSP-Projektpraktikum" adressiert Forschungsthemen im Bereich von distributed systems (DS, deutsch verteilten Anwendungen) und Programmiersprachen für DS. Die angebotenen Themen hängen von der aktuellen Forschung der DSP Gruppe ab und umfassen unter anderem: <ul style="list-style-type: none"><li>• Software-defined networking (SDN)</li><li>• Network function virtualization (NFV) and in-network processing (INP)</li><li>• Traffic engineering (TE)</li><li>• Network monitoring</li><li>• Resource management in datacenters (RMF)</li><li>• Big data analytics (Spark, YARN, OpenStack, ..)</li><li>• Event-based systems</li><li>• Security in SDN, INP, and big data</li><li>• Geo-distributed data processing</li><li>• Compiler infrastructures for DS</li><li>• Language abstractions for DS</li><li>• Session types / calculi for DS</li><li>• Network Protocols</li></ul> Die teilnehmenden Studierenden realisieren ein Forschungsprojekt welches zusammen mit den Betreuern definiert wird. Das "DSP: Projektpraktikum" hat im Vergleich zum "DSP: Praktikum" einen größeren Umfang.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach der Teilnahme am "DSP-Projektpraktikum" können Studierende technische und wissenschaftliche Probleme im Bereich DS lösen.  Je nach ausgewähltem Thema erlernen Studierende folgende Kompetenzen:  Entwurf komplexer DS Methodische Analyse und Auswertung von:				



	<ul style="list-style-type: none"> <li>• Modellen</li> <li>• Experimenten</li> <li>• Software</li> <li>• Entwurf von Programmiersprachen</li> <li>• Schreiben von technischen Dokumenten oder Projektberichten</li> <li>• Erstellen und vortragen eines Abschlussvortrages</li> </ul>
4	<p><b>Voraussetzung für die Teilnahme</b>  Empfohlen: Interesse am Erarbeiten von Lösungsvorschlägen für herausfordernde Probleme im Bereich DS, eigenverantwortliches arbeiten und ein großes Interesse an aktuellen Forschungsthemen.</p> <p>Da die angebotenen Themen ein großes Themengebiet abdecken, sind die Anforderungen sehr verschieden und projektabhängig. Eine detaillierte Beschreibung der Themen als auch der Anforderungen wird während des ersten Termins präsentiert.</p>
5	<p><b>Prüfungsform</b>  Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0984-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b>  Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b>  Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0984-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b>  B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Fortgeschrittene Themen in Eingebetteten Systemen und ihren Anwendungen					
<b>Modul Nr.</b> 20-00-1001	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1001-pp	Fortgeschrittene Themen in Eingebetteten Systemen und ihren Anwendungen	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> Der Kurs bearbeitet aktuelle Forschungs- und Entwicklungsthemen aus dem Bereich von Rechnersystemen und Programmierwerkzeugen, auch speziell im Umfeld von eingebetteten und anwendungsspezifischen Architekturen. Die Themen bestimmen sich aus den spezifischen Arbeitsgebieten der Mitarbeiter und vermitteln technische und einleitende wissenschaftliche Kompetenzen, zum Beispiel aus einem oder mehreren der folgenden Gebiete:  <ul style="list-style-type: none"><li>- Rechnerarchitekturen auf Prozessor- und Systemebene</li><li>- Entwurf digitaler Schaltungen und Hardware-Systeme</li><li>- Einsatz von Field-Programmable Gate Arrays</li><li>- Hardware/Software-Entwurfs- und Programmierwerkzeuge</li><li>- Betriebssysteme und hardware-nahe Programmierung</li><li>- Hardware/Software-Co-Design</li><li>- Anwendungsspezifische Architekturen und Techniken</li><li>- Entwurf und/oder Programmierung von Rechenbeschleunigern</li><li>- Debugging und Analyseverfahren für Hardware/Software-Systeme</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Der/die Studierende sollen Erfahrungen mit der Einarbeitung in ein neues Themenfeld und der praktischen Bearbeitung einer komplexeren Aufgabe aus diesem sammeln. Zu diesen Erfahrungen können Literaturrecherchen, das Einarbeiten in bestehende Code-Basen aus dem Hardware/Software-Bereich, sowie ganz praktische Implementierung von Hardware und/oder Software gehören. Beim Abschlussvortrag sind auch geeignete Präsentationstechniken anzuwenden.				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: <p>Das Interesse, zu den Lehrinhalten anspruchsvolle Lösungen zu entwickeln. Dabei sind jeweils themenspezifische Kenntnisse, u.a. zum Hardware-Entwurf, dem Compilerbau und der parallelen Programmierung erforderlich. Diese Kenntnisse können beispielsweise durch den Besuch der entsprechenden Lehrveranstaltungen erworben werden.</p>
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1001-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)</p>
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1001-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Parallele Programmiertechnologie					
<b>Modul Nr.</b> 20-00-1008	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1008-pr	Parallele Programmiertechnologie	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Die Praktikumssteilnehmer entwickeln Technologien zur parallelen Programmierung aus den folgenden Themenbereichen und/oder wenden diese an: <ul style="list-style-type: none"><li>• Erschließung möglicher Parallelität</li><li>• Leistungsanalyse und –modellierung</li><li>• Korrektheitsanalyse</li><li>• Profiling</li><li>• Skalierbare Algorithmen</li><li>• Ressourcenmanagement und Scheduling</li><li>• Anwendungen (z.B. Deep Learning)</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <ul style="list-style-type: none"><li>• Kennenlernen und praktische Entwicklung und/oder Anwendung paralleler Programmiertechnologien</li><li>• Einüben softwaretechnischer Methoden</li><li>• Teamarbeit in Softwareprojekten</li><li>• Präsentation von Projektergebnissen in Berichten und Vorträgen</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Kenntnisse paralleler Programmierung und Systeme				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-1008-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1008-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname Compiler Tooling					
Modul Nr. 20-00-1013	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1013-pr	Compiler Tooling	6	Praktikum	4
2	<b>Lerninhalt</b> Moderne Compiler zielen primär darauf ab, effizienten Code für eine bestimmte Plattform zu generieren und hierfür nutzen sie fortgeschrittene Analysis- und Transformationswerkzeuge. Eine solche Infrastruktur ist aber auch nützlich für Quellcodetransformation, z.B. für Werkzeuge, die Codes annotieren, instrumentieren, oder in eine kanonische Form bringen. Die Entwicklung solcher Werkzeuge ist für die C++ Sprache aufgrund ihrer Komplexität eine Herausforderung. Eine offene Compiler Infrastruktur, die in einer Vielzahl von Forschungs- und Produktionscompilern genutzt wird, ist die LLVM Infrastruktur ( <a href="http://www.llvm.org">www.llvm.org</a> ). Ein vielgenutztes Front-End für C, C++ und objective C ist Clang, welches mächtige Mechanismen für die Extraktion von Information aus dem abstrakten Syntaxbaum zur Verfügung stellt, und so Modifikationen des Quellcodes wie auch die Generierung der Zwischenrepräsentation von LLVM ermöglicht.  Die Studierenden arbeiten mit verschiedenen Komponenten und Techniken des Clang/LLVM Frameworks und implementieren praktische Übungen für Quelltransformationen. Die Clang/LLVM Techniken beinhalten insbesondere die Handhabung und Matching Techniken auf dem abstrakten Syntaxbaum von Clang. Beispiele von Quelltransformationen werden verschiedene Facetten von Code-Erweiterung und -Refactoring beinhalten, z.B. für die Instrumentierung paralleler Codes, für die Übermittlung von Information zwischen der statischen Analyse und der Laufzeitumgebung von (parallelen) Codes, oder für Code Refactoring um bestimmte Coding Standards einzuhalten.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem Besuch dieses Kurses kennen die Studierenden grundlegende und fortgeschrittene Konzepte der syntaktischen und semantischen Code Analyse und Quelltext-Transformation, basierend auf der Clang/LLVM Technologie. Insbesondere können sie auf spezielle Aufgaben zugeschnittene statische Analyse- und Code- Transformations Werkzeuge entwerfen und implementieren, das geeignete Abstraktionsniveau für die zu lösende Aufgabe reflektieren und entscheiden, und weitere Nutzungsszenarien für Compiler Technologie erstellen.				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch von „Einführung in Compilerbau“ und „Parallele Programmierung“ oder vergleichbarer Veranstaltungen, Kenntnisse von C++
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1013-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1013-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Data Management - Praktikum					
<b>Modul Nr.</b> 20-00-1041	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1041-pr	Data Management - Praktikum	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Die Teilnehmenden lösen in kleinen Projektgruppen ein gegebenes Problem. Bei den Problemen handelt es sich um Programmierprojekte, die sich auf Fragestellungen aus aktuellen Forschungsthemen des Data Management Lab beziehen.  Mögliche Themenbereiche sind: - Skalierbare Datenbanksysteme und moderne Hardware - Cloud Datenbanken und Blockchains - Interaktive Daten- und Textexploration - Natural Language Interfaces für Datenbanken - Skalierbare Systeme für Maschinelles Lernen  In dieser Veranstaltung setzen Studierende ein ausgewähltes Projekt um. Im Vergleich zum Praktikum haben die Probleme des Projektpraktikums einen erweiterten Umfang.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Beendigung der Veranstaltung haben Studierende folgende Lernziele erreicht: - Vertieftes Verständnis von aktuellen Techniken für moderne Datenmanagement-Systeme - Anwendung und Implementierung der Techniken in individuellen Projekten - Evaluierung von möglichen Designalternativen mit Hilfe von Benchmarks bzw. realen Workloads				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Abhängig vom ausgewählten Thema.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1041-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				



	Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1041-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Data Management - Projektpraktikum					
<b>Modul Nr.</b> 20-00-1042	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1042-pp	Data Management - Projektpraktikum	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> Die Teilnehmenden lösen in kleinen Projektgruppen ein gegebenes Problem. Bei den Problemen handelt es sich um Programmierprojekte, die sich auf Fragestellungen aus aktuellen Forschungsthemen des Data Management Lab beziehen.  Mögliche Themenbereiche sind: - Skalierbare Datenbanksysteme und moderne Hardware - Cloud Datenbanken und Blockchains - Interaktive Daten- und Textexploration - Natural Language Interfaces für Datenbanken - Skalierbare Systeme für Maschinelles Lernen  In dieser Veranstaltung setzen Studierende ein ausgewähltes Projekt um. Im Vergleich zum Praktikum haben die Probleme des Projektpraktikums einen erweiterten Umfang.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Beendigung der Veranstaltung haben Studierende folgende Lernziele erreicht: - Vertieftes Verständnis von aktuellen Techniken für moderne Datenmanagement-Systeme - Anwendung und Implementierung der Techniken in individuellen Projekten - Evaluierung von möglichen Designalternativen mit Hilfe von Benchmarks bzw. realen Workloads				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Abhängig vom ausgewählten Thema.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1042-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1042-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**  
**Vertiefung Distributed Computing**  
**Wahlbereich Studienbegleitende Leistungen**  
**Seminare**

## Modulbeschreibung

<b>Modulname</b> Forschungsseminar zu Netzen, Sicherheit, Mobilität und Drahtloser Kommunikation					
<b>Modul Nr.</b> 20-00-0549	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 120 h	<b>Selbststudium</b> 75 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0549-se	Forschungsseminar zu Netzen, Sicherheit, Mobilität und Drahtloser Kommunikation	4	Seminar	3
<b>2</b>	<b>Lerninhalt</b> Das Forschungsseminar zu Netzen, Sicherheit, Mobilität und Drahtloser Kommunikation erarbeitet aktuelle Fragstellungen, die als hoch-relevant für die zukünftige Entwicklung der genannten Themenfelder eingeschätzt werden. Es umfasst das Studium, die kritische Analyse und Diskussion, das Zusammenfassen und die Präsentation ausgewählter erstklassiger Forschungsbeiträge. Ein Einblick in wissenschaftliche Arbeitsweise wird vermittelt. Ein Kurzreferat und ein abschließendes Referat sowie eine schriftliche Ausarbeitung werden erstellt. Die Themen des Forschungsseminars speisen sich aus den aktuellen Forschungsthemen der Arbeitsgruppe SEEMOO.  Lernziele: - Eigenständiges Einarbeiten in ein Thema auf dem Gebiet Kommunikationsnetze, Sicherheit, Mobilität und Drahtloser Kommunikation (i.d.R. englischsprachig) - Eigene darüber hinausgehende Literaturrecherchen - Interpretation und Einordnen der Ergebnisse der Literaturarbeit - Erstellen eines einführenden und eines vertiefenden Vortrags über die Thematik einschließlich Folienpräsentationen - Halten der beiden Vorträge vor einem Publikum mit heterogenem Vorwissen - Fachdiskussion nach jedem Vortrag - Feedback an die Vortragenden zu den Vorträgen (u.a. betreffend Rhetorik, Präsentationstechniken) und zur Fachdiskussion - Kennen des wissenschaftlichen Arbeitsprozesses und Publikationsprozesses				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung besitzen die Studierenden die Fähigkeit selbstständig wissenschaftlich neue Themen zu erschließen. Sie haben ein tiefgreifendes Verständnis ausgewählter Basismechanismen, Methoden und Anwendungen in dem				

	bearbeiteten Themenfeld erworben. Arbeitstechniken wie ausführliche Literaturrecherche, kritische Diskussion und Analyse wissenschaftlicher Artikel und die Presentation der erzielten Arbeitsergebnisse werden von den Studierenden beherrscht. Die Studierenden können ihre Arbeit vor einem kritischen Fachpublikum verteidigen.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreiche Teilnahme an einer integrierten Veranstaltung des Fachgebiets SEEMOO
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0549-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0549-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Themenspezifisch ausgewählte, aktuelle wissenschaftliche Veröffentlichungen
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Seminar zu Netzen, Sicherheit, Mobilität und Drahtloser Kommunikation					
<b>Modul Nr.</b> 20-00-0582	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0582-se	Seminar zu Netzen, Sicherheit, Mobilität und Drahtloser Kommunikation	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Das Seminar zu Netzen, Sicherheit, Mobilität und Drahtloser Kommunikation erarbeitet aktuelle Fragestellungen auf den genannten Gebieten. Unter Anleitung der Dozenten umfasst es das Studium, die kritische Analyse und Diskussion, das Zusammenfassen und die Präsentation ausgewählter Forschungsbeiträge. Ein Kurzreferat und ein abschließendes Referat sowie eine schriftliche Ausarbeitung werden erstellt. Die Themen des Seminars speisen sich aus den aktuellen Forschungsthemen der Arbeitsgruppe SEEMOO. Lernziele: - Eigenständiges Einarbeiten in ein Thema auf dem Gebiet Kommunikationsnetze, Sicherheit, Mobilität und Drahtloser Kommunikation (i.d.R. englischsprachig) - Darüber hinausgehende Literaturrecherchen, angeleitet von Betreuer - Interpretation und Einordnen der Ergebnisse der Literaturarbeit, angeleitet von Betreuer - Erstellen eines einführenden und eines vertiefenden Vortrags über die Thematik einschließlich Folienpräsentationen, angeleitet von Betreuer - Halten der beiden Vorträge vor einem Publikum mit heterogenem Vorwissen - Fachdiskussion nach jedem Vortrag - Feedback an die Vortragenden zu den Vorträgen (u.a. betreffend Rhetorik, Präsentationstechniken) und zur Fachdiskussion				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung besitzen die Studierenden die Fähigkeit unter Anleitung wissenschaftlich zu arbeiten. Sie kennen die grundlegenden Techniken der wissenschaftlichen Literaturarbeit und können diese für ein definiertes Thema anwenden. Sie haben ein mitteltiefes Verständnis ausgewählter Basismechanismen, Methoden und Anwendungen in dem bearbeiteten Themenfeld. Die Studierenden können dieses erworbene				

	Wissen einem heterogenen Publikum verständlich präsentieren und die technischen Details des bearbeiteten Themas erläutern.
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreiche Teilnahme an einer Integrierten Veranstaltung des Fachgebiets SEEMOO
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0582-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0582-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Themenspezifisch ausgewählte, aktuelle wissenschaftliche Veröffentlichungen
10	<b>Kommentar</b>



## Modulbeschreibung

Modulname					
Symbolische Ausführung					
Modul Nr. 20-00-0702	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0702-se	Symbolische Ausführung	3	Seminar	2
2	<b>Lerninhalt</b> Symbolische Ausführung von Programmen ist eine fundamentale Analysetechnik, die u.a. die Basis von Testgenerierung, Compileroptimierung, Verifikation oder Visualisierung darstellt. In den letzten Jahren wurden darin bedeutende Fortschritte erzielt. Im Seminar werden die wichtigsten klassischen und neuen Arbeiten zur symbolischen Ausführung vorgestellt.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach Besuch der Veranstaltung verstehen die Teilnehmenden, was die Möglichkeiten und Grenzen dieser fundamentalen Programmanalysetechnik sind.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0702-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0702-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik				

	M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Aktuelle Themen zu Nebenläufigkeit und Parallelität					
<b>Modul Nr.</b> 20-00-0960	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0960-se	Aktuelle Themen zu Nebenläufigkeit und Parallelität	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> In diesem Seminar werden Forschungsartikel zu verschiedenen Aspekten von Nebenläufigkeit und Parallelität diskutiert; die Forschungsartikel behandeln beispielsweise:  - Semantik der Nebenläufigkeit (Interleaving-Sematik, Multicore-Semantik, Weak Memory Models), - Parallele Architekturen (Grundlagen von parallelen Architekturen, symmetrische Multiprozessorsysteme, Massenparallelrechner), - Parallele Programmierung (parallele Programmierungsmodelle, Kommunikation, Synchronisation), - Parallelisierung und Kompilierung (Voll-/Halbautomatische Parallelisierung, Datenabhängigkeiten, Lastverteilung), - Verifikation von nebenläufigen Programmen (Separation Logic, Rely/Guarantee Reasoning).				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an diesem Seminar werden die Studierenden fähig sein, aktuelle Entwicklungen in den Bereichen Nebenläufigkeit und Parallelität zu diskutieren. Des Weiteren werden die Studierenden ihre Fähigkeiten im Lesen und Verstehen wissenschaftlicher Artikel und im Präsentieren, Diskutieren und Vergleichen wissenschaftlicher Ergebnisse verbessern.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Informatikkenntnisse entsprechend der ersten vier Semester des Bachelorstudiengangs Informatik.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0960-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0960-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Schutz von verteilten Infrastrukturen und Netzwerken					
<b>Modul Nr.</b> 20-00-1022	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1022-se	Schutz von verteilten Infrastrukturen und Netzwerken	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Das Seminar zum Schutz von verteilten Infrastrukturen und Netzwerken setzt sich aus der strukturierten Arbeit an wissenschaftlichen Veröffentlichungen auseinander. Die Themen befassen sich hierbei mit: - Vertrauen - Privatheit - Resilienz in Infrastrukturen und Netzwerken.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studenten, die an dem Seminar teilnehmen, haben die Chance die Themen durch strukturierte Forschung, näher kennen zu lernen. Ihre Aufgabe wird es sein, aktuelle wissenschaftliche Veröffentlichungen zu verstehen, um deren Beitrag zu erklären. Außerdem muss ein Survey über das bearbeitete Thema verfasst werden.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Grundlegendes Verständnis von IT-Sicherheit und verteilten Systemen.  Veranstaltungen: Computersystemsicherheit Computer Netze und verteilte Systeme				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1022-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1022-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Erweitertes Seminar - Systems and Machine Learning					
<b>Modul Nr.</b> 20-00-1057	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 120 h	<b>Selbststudium</b> 75 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1057-se	Erweitertes Seminar - Systems and Machine Learning	4	Seminar	3
<b>2</b>	<b>Lerninhalt</b> <p>Dieses Seminar dient der Diskussion neuer Forschungspapiere im Zusammenhang von Hardware-/Softwaresystemen und maschinellem Lernen (ML). Das Seminar zielt auf die Verbindungen zwischen diesen Themenbereichen ab und diskutiert Fragestellungen, die auf praktisch anwendbares maschinelles Lernen zugeschnitten sind wie z.B. Hardware-Beschleuniger für ML, verteilte skalierbare ML-Systeme, neuer Programmierparadigmen für ML, Automatisiertes ML, sowie Anwendungen von ML für Systeme.</p> <p>Jeder Teilnehmer/jede Teilnehmerin präsentiert ein Forschungspapier, das anschließend von allen Teilnehmenden diskutiert wird. Darüber hinaus werden zusammenfassende Arbeiten in Gruppen verfasst und einem Peer-Review Prozess unterzogen. Die vorzustellenden Arbeiten stellen in der Regel aktuelle Publikationen in relevanten Konferenzen und Zeitschriften dar.</p> <p>Das Seminar wird als Blockveranstaltung angeboten.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach diesem Seminar sind Studierende in der Lage:</p> <ul style="list-style-type: none"><li>- einen unbekannten Text aus den Bereichen des Seminars selbständig aufzuarbeiten</li><li>- eine Präsentation und eine schriftliche Zusammenfassung für ein Fachpublikum in diesem Gebiet zu entwickeln</li><li>- an einer Fachdiskussion über ein Thema aus den Bereichen des Seminars sinnvoll teilzunehmen</li><li>- die Meinung über eine wissenschaftliche Arbeit in der Form eines schriftlichen Peer-Reviews zu artikulieren</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Grundkenntnisse in maschinellem Lernen, skalierbarem Datenmanagement und Hardware-/Softwaresystemen.</p>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1057-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1057-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b>					
Distributed Systems Programming: Seminar					
<b>Modul Nr.</b> 20-00-1066	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1066-se	Distributed Systems Programming: Seminar	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Das "DSP-Praktikum" adressiert Forschungsthemen im Bereich von distributed systems (DS, deutsch verteilten Anwendungen) und Programmiersprachen für DS. Die angebotenen Themen hängen von der aktuellen Forschung der DSP Gruppe ab und umfassen unter anderem: - Software-defined networking (SDN) - Network function virtualization (NFV) and in-network processing (INP) - Traffic engineering (TE) - Network monitoring - Resource management in datacenters (RMF) - Big data analytics (Spark, YARN, OpenStack, ..) - Event-based systems - Security in SDN, INP, and big data - Geo-distributed data processing - Compiler infrastructures for DS - Language abstractions for DS - Session types / calculi for DS - Network Protocols  Die teilnehmenden Studierenden realisieren eine Seminararbeit welches zusammen mit dem Betreuer definiert wird.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach der Teilnahme am "DSP-Seminar" können Studierende technische und wissenschaftliche Probleme im Bereich DS aufarbeiten und präsentieren.  Je nach ausgewähltem Thema erlernen Studierende folgende Kompetenzen: - Literaturrecherche im Themenbereich - Methodische Analyse und Auswertung von: - Modellen - Experimenten - Software				

	<ul style="list-style-type: none"> <li>- Schreiben von technischen Dokumenten oder Projektberichten</li> <li>- Erstellen und Vortragen eines Abschlussvortrages</li> </ul>
4	<p><b>Voraussetzung für die Teilnahme</b></p> <p>Empfohlen:</p> <p>Interesse am Erarbeiten von Lösungsvorschlägen für herausfordernde Probleme im Bereich DS, eigenverantwortliches arbeiten und ein großes Interesse an aktuellen Forschungsthemen.</p> <p>Vorlesung TK1 (optional)</p> <p>Da die angebotenen Themen ein großes Themengebiet abdecken, sind die Anforderungen sehr verschieden und projektabhängig. Eine detaillierte Beschreibung der Themen als auch der Anforderungen wird in der ersten Vorlesung präsentiert und können anschließend von den Studenten ausgewählt werden.</p>
5	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1066-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1066-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Aktor-basierte Programmiersprachen					
<b>Modul Nr.</b> 20-00-1074	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1074-se	Aktor-basierte Programmiersprachen	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Im Zentrum des Seminars stehen Aktor-basierte Modellierungs- und Programmiersprachen wie Scala/Akka, ABS, Encore, u.ä.  Teilnehmer_innen dieses Seminars sollen einzelne Vertreter der Aktor-basierten Sprachen vorstellen, die realisierten Konzepte erklären und diskutieren.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> * Fähigkeit ein wissenschaftliche Thema aufzuarbeiten und zu präsentieren * Fähigkeit wissenschaftliche Berichte zu lesen und verwandte Arbeiten zu recherchieren * Erwerb von Wissen über Aktor-basierte Sprachen und deren Anwendung				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Interesse in Programmiersprachen und verteilten Systemen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1074-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>[20-00-1074-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Fortgeschrittene Techniken der Softwareverifikation					
<b>Modul Nr.</b> 20-00-1078	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1078-se	Fortgeschrittene Techniken der Softwareverifikation	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Im Seminar befassen Sie sich mit Themen zu den aktuellen Forschungsinhalten der Arbeitsgruppe Semantik und Verifikation paralleler System. Es werden sowohl klassische als auch aktuelle Forschungsarbeiten im Bereich Softwareverifikation (d.h. Model Checking, Programmanalyse, Testen, etc.) behandelt.  Während des Seminars werden Sie unter Anleitung - sich auf Basis von vorgegebener und selbst gefundener, wissenschaftlicher Literatur in Ihr Thema einarbeiten - einen Vortrag über Ihr Thema vorbereiten und vor den anderen Teilnehmern halten, um mit ihnen anschließend über Ihr Thema zu diskutieren, - eine wissenschaftliche Ausarbeitung verfassen, die einen zusammenfassenden Überblick über Ihr Thema gibt.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Abschluss des Seminars können sich die Studierenden anhand von Ausgangsliteratur eigenständig in ein wissenschaftliches Thema einarbeiten und dieses Thema einem heterogenen Fachpublikum sowohl mündlich als auch schriftlich präsentieren.  Im Detail können die Studierenden Methoden zur Literaturrecherche anwenden und die Relevanz von gefundener Literatur beurteilen. Sie können den wesentlichen Inhalt einer wissenschaftlichen Veröffentlichung ermitteln und diesen kritisch beurteilen. Außerdem sind sie in der Lage verschiedene wissenschaftliche Arbeiten miteinander zu vergleichen. In einem mündlichen Vortrag können die Studierenden ihr Thema und ihre Ergebnisse einem heterogenen Fachpublikum erklären und ihre Ergebnisse vor diesem Publikum verteidigen. Zusätzlich können die Studierenden in einer schriftlichen Ausarbeitung ihr Thema und ihre Ergebnisse beschreiben.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b>				

	<p>Empfohlen: Informatik- und Mathematikkenntnisse entsprechend den ersten 4 Semestern des Bachelorstudiengangs Informatik</p> <p>Hilfreich: Besuch einer Veranstaltung des Fachgebietes Semantik und Verifikation paralleler Systeme</p>
<b>5</b>	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1078-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1078-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Aktuelle Themen aus dem Bereich Concurrency Theory					
<b>Modul Nr.</b> 20-00-1093	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1093-se	Aktuelle Themen aus dem Bereich Concurrency Theory	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Moderne Software-Systeme sind verteilt und Abhängig von Kommunikation. Dies führt zu zusätzlichen Problem bei der Verifikation solcher Systeme, mit denen sich das Forschungsgebiet Concurrency Theory beschäftigt. Hier werden verschiedene Methoden zur Modellierung, Simulation und Analyse verteilter Systeme untersucht. Der Bereich der Modellierungssprachen umfasst graphische Modelle so wie Petrinetze oder Event Structures genauso wie Programiersprachen nahe Modelle wie z.B. Prozesskalküle. Um solche Systeme zu analysieren, wurden verschiedene Techniken so wie Typsysteme, Model Checking und interaktives Theorembeweisen auf die speziellen Ansprüche verteilter Systeme angepasst und viele neue Techniken wurden entwickelt. In diesem Seminar werden aktuelle Forschungspapier aus dem Bereich Concurrency Theory vorgestellt und diskutiert.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme haben Studierende eine Vorstellung von den aktuell bearbeiteten Themen im Bereich Concurrency Theory. Sie können Resultate aus diesem Bereich präsentieren und die Vorteile der präsentierten Methoden kritisch hinterfragen. Die Vorstellung der vorliegenden Resultate wird ihnen dabei helfen ihre eigenen Arbeiten klar darzustellen und zu verteidigen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen werden Informatik- und Mathematikkenntnisse entsprechend den ersten 4 Semestern des Bachelorstudiengangs Informatik, insbesondere formalen Sprachen.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1093-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Kolloquium (optional: einschließlich Präsentation), Hausarbeit
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1093-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

Modulname					
Seminar Softwaresystemtechnologie					
Modul Nr. 18-su-2080	Leistungspunkte 4 CP	Arbeitsaufwand 120 h	Selbststudium 90 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. rer. nat. Andreas Schürr		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	18-su-2080-se	Seminar Softwaresystemtechnologie	0	Seminar	2
2	<b>Lerninhalt</b> In diesem Seminar werden von den Studierenden wissenschaftliche Ausarbeitungen aus wechselnden Themenbereichen angefertigt. Dies umfasst die Einarbeitung in ein aktuelles Thema der IT-Systementwicklung mit schriftlicher Präsentation in Form einer Ausarbeitung und mündlicher Präsentation in Form eines Vortrages. Die Themen des aktuellen Semesters sind der Webseite der Lehrveranstaltung zu entnehmen <a href="https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/sst-s">https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/sst-s</a> .				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage sich in ein unbekanntes Themengebiet einzuarbeiten, die Zuverlässigkeit von Informationsquellen einzuschätzen und diese nach wissenschaftlichen Aspekten aufzuarbeiten. Studierende erlernen die Bearbeitung eines Themas durch Literaturrecherche zu unterstützen und kritisch zu hinterfragen. Weiterhin wird die Fähigkeit erworben, ein klar umrissenes Thema in Form einer schriftlichen Ausarbeitung und in Form eines mündlichen Vortrags unter Anwendung von Präsentationstechniken zu präsentieren.				
4	<b>Voraussetzung für die Teilnahme</b> Grundkenntnisse der Softwaretechnik sowie Programmiersprachenkenntnisse				
5	<b>Prüfungsform</b> Modulabschlussprüfung: <ul style="list-style-type: none"><li>• Modulprüfung (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Bericht und/oder Präsentation und/oder Kolloquium. Die Art der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben.				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				
7	<b>Benotung</b>				

	Modulabschlussprüfung: <ul style="list-style-type: none"> <li>• Modulprüfung (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> BSc iST, BSc Informatik, MSc ETiT
<b>9</b>	<b>Literatur</b> <a href="https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/sst-s">https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/sst-s</a>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Distributed Computing**

**Wahlbereich Studienbegleitende Leistungen**

**Praktikum in der Lehre**

## Modulbeschreibung

Modulname					
Praktikum in der Lehre - Formale Methoden im Softwareentwurf					
Modul Nr. 20-00-0531	Leistungspunkte 5 CP	Arbeitsaufwand 150 h	Selbststudium 105 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0531-pl	Praktikum in der Lehre - Formale Methoden im Softwareentwurf	5	Praktikum in der Lehre	3
2	Lerninhalt Vorbereitung und Korrektur von Übungsaufgaben, Betreuung von Übungsgruppen				
3	Qualifikationsziele / Lernergebnisse Es sollen Fähigkeiten erlernt werden, geeignete Lernmaterialien für Schulungen in Informatikthemen selbst zu erstellen, ihren Einsatz kritisch zu begleiten und dabei auch die Lernenden zu betreuen und anzuleiten.				
4	Voraussetzung für die Teilnahme Empfohlen: Formale Methoden im Softwareentwurf				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0531-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%)				
7	Benotung Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0531-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Internetsicherheit und Sicherheit in Mobilen Netzen					
<b>Modul Nr.</b> 20-00-0957	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0957-pl	Praktikum in der Lehre - Internetsicherheit und Sicherheit in Mobilen Netzen	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> <p>Dieser Kurs befasst sich mit damit Lehrinhalte der Themenschwerpunkte Internetsicherheit und Sicherheit in Mobilen Netzen didaktisch aufzubereiten und durch begleitende praktische Übungen besser verständlich zu machen.</p> <p>Dies umfasst unter anderem: Die Implementierung von Systemen die in der Vorlesung behandelte Schwachstellen aufweisen und den Studierenden für praktische Übungen verfügbar gemacht werden; die Erstellung von Minitests zur Leistungskontrolle; die Konzeption von Materialien für leistungsschwache wie leistungsstarke Studenten um Inhalte der Vorlesung zu vertiefen; das Erstellen von anspruchsvollen Bonussystemen.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Die Studierenden können nach erfolgreicher Durchführung der Veranstaltung:</p> <ul style="list-style-type: none"><li>- Lehrinhalte aus der Vorlesung für Haus- und Präsenzübungen aufbereiten</li><li>- Praxisnahe Übungsformen konzipieren und erstellen</li><li>- Übungen mit Studierendengruppen aller Leistungsniveaus konzipieren und durchführen</li><li>- Ein Konzept für aufeinander aufbauende praktische Übungen entwickeln</li><li>- Methoden der Lernkontrolle für die Lerninhalte der Vorlesung anwenden</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Erfolgreicher Besuch der SEEMOO Veranstaltung für die das PIDL durchgeführt wird.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0957-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0957-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Praktikum in der Lehre - Einführung in den Compilerbau					
<b>Modul Nr.</b> 20-00-0988	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0988-pl	Praktikum in der Lehre - Einführung in den Compilerbau	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> - Erstellen von Übungs- und Lehrmaterial zu Einführung in den Compilerbau - Betreuung von Studierenden zu Themen der Einführung in den Compilerbau, insbesondere unter Verwendung des neuen Lehrmaterials				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Absolvieren der Veranstaltung sind die Studierenden in der Lage, selbständig Lehrmaterialien zu Informatikthemen zu erstellen. Sie können das Material in Schulungen erfolgreich einsetzen und seine didaktische Wirksamkeit kritisch beurteilen. Sie können Studierende in direktem persönlichen Kontakt, aber auch über elektronische Kommunikationsmedien anleiten und betreuen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Kenntnisse aus Funktionale und Objektorientierte Programmierkonzepte, Algorithmen und Datenstrukturen, Einführung in den Compilerbau sowie Rechnerorganisation (oder vergleichbaren Veranstaltungen)				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0988-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				



	<ul style="list-style-type: none"> <li>[20-00-0988-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Data Management					
<b>Modul Nr.</b> 20-00-1040	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1040-pl	Praktikum in der Lehre - Data Management	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Erstellung von Übungs- und Vorlesungsmaterial				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Erfahrung in der Betreuung von Studierenden im Themenbereich Datenmanagement, mit dem Fokus auf das neu erstellte Übungs- und Vorlesungsmaterial				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch der Veranstaltung „Informationsmanagement“ oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1040-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1040-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum in der Lehre - Echtzeitsysteme					
<b>Modul Nr.</b> 20-00-1060	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Distributed Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1060-pl	Praktikum in der Lehre - Echtzeitsysteme	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Konzeption, Betreuung und Durchführung von Übungen sowie vorlesungsbegleitenden Praktika der Vorlesung „Echtzeitsysteme“.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende sind in der Lage: <ul style="list-style-type: none"><li>• Lehrinhalte in Übungen zu präsentieren und zu erklären</li><li>• Praktikumsgruppen zu betreuen</li><li>• Methoden zur Kontrolle des Lernerfolgs systematisch anzuwenden</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Erfolgreiche Absolvierung der Echtzeitsysteme-Veranstaltung oder entsprechende Kenntnisse.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-1060-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-1060-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**  
**Vertiefung Visual Computing**  
**Wahlbereich Computer Graphics**

## Modulbeschreibung

<b>Modulname</b> Graphische Datenverarbeitung I					
<b>Modul Nr.</b> 20-00-0040	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0040-iv	Graphische Datenverarbeitung I	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Einführung in die Grundlagen der Computergraphik, insb. Ein- u. Ausgabegeräte, Rendering Pipeline am Beispiel von OpenGL, räumliche Datenstrukturen, Beleuchtungsmodelle, Ray Tracing, aktuelle Entwicklungen in der Computergraphik				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch dieser Veranstaltung sind Studierende in der Lage alle Komponenten der Graphikpipeline zu verstehen und dadurch variable Bestandteile (Vertex-Shader, Fragment-Shader, etc.) anzupassen. Sie können Objekte im 3D-Raum anordnen, verändern und effektiv speichern, sowie die Kamera und die Perspektive entsprechend wählen und verschiedene Shading-Techniken und Beleuchtungsmodelle nutzen, um alle Schritte auf dem Weg zum dargestellten 2D-Bild anzupassen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: <ul style="list-style-type: none"><li>• Programmierkenntnisse</li><li>• Kenntnisse über grundlegende Algorithmen und Datenstrukturen</li><li>• Kenntnisse im Bereich Lineare Algebra</li><li>• Kenntnisse im Bereich Analysis</li><li>• Inhalte der Vorlesung „Visual Computing“ oder einer vergleichbaren Veranstaltung</li></ul>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0040-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0040-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M.Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <ul style="list-style-type: none"> <li>• Real-Time Rendering: Tomas Akenine-Möller, Eric Haines, Naty Hoffman A.K. Peters Ltd., 3rd edition, ISBN 987-1-56881-424-7</li> <li>• Fundamentals of Computer Graphics: Peter Shirley, Steve Marschner, third edition, ISBN 979-1-56881-469-8</li> <li>• Weitere aktuelle Literaturhinweise werden in der Veranstaltung gegeben.</li> </ul>
10	<p><b>Kommentar</b></p>



## Modulbeschreibung

<b>Modulname</b> Graphische Datenverarbeitung II					
<b>Modul Nr.</b> 20-00-0041	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0041-iv	Graphische Datenverarbeitung II	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Grundlagen der verschiedenen Objekt- und Oberflächen-Repräsentationen in der graphischen Datenverarbeitung. Kurven und Oberflächen (Polynome, Splines, RBF) Interpolation und Approximation, Displaytechniken, Algorithmen: de Casteljau, de Boor, Oslo, etc. Volumen und implizite Oberflächen. Visualisierungstechniken, Iso-Surfaces, MLS, Oberflächen-Rendering, Marching-Cubes. Polygonnetze. Netz Kompression , Netz-Vereinfachung, Multiskalen Darstellung, Subdivision. Punktwolken: Renderingtechniken, Oberflächen-Rekonstruktion, Voronoi-Diagramme und Delaunay-Triangulierung.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch dieser Veranstaltung sind Studierende in der Lage mit diversen Objekt- und Oberflächen-Repräsentationen umzugehen, das heißt diese zu verwenden, anzupassen, anzuzeigen (rendern) und effektiv zu speichern. Dazu gehören mathematisch polynomiale Repräsentationen, Iso-oberflächen, volumen Darstellungen, implizite Oberflächen, Polygonnetze, Subdivision-Kontrollnetze und Punktwolken.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Der vorherige Besuch von „Algorithmen und Datenstrukturen“ und „Graphische Datenverarbeitung I“ oder vergleichbaren Veranstaltungen  Kenntnisse über Grundlagen aus der Höheren Mathematik  Programmierkenntnisse in C / C++				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0041-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0041-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <ul style="list-style-type: none"> <li>• Real-Time Rendering: Tomas Akenine-Möller, Eric Haines, Naty Hoffman A.K. Peters Ltd., 3rd edition, ISBN 987-1-56881-424-7</li> <li>• Weitere aktuelle Literaturhinweise werden in der Veranstaltung gegeben.</li> </ul>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Geometrische Methoden des CAE/CAD					
<b>Modul Nr.</b> 20-00-0140	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0140-iv	Geometrische Methoden des CAE/CAD	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• parametrische Kurvenmodelle</li><li>• parametrische Flächenmodelle</li><li>• Topologie und CAD-Volumenmodelle</li><li>• CAD-Operationen auf Flächen</li><li>• Tessellierung</li><li>• Approximation von Kurven und Flächen</li><li>• Finite-Elemente-Methode und Strömungssimulation</li><li>• verschiedene Anwendungen aus dem CAD-Bereich</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Studierende beherrschen nach erfolgreichem Besuch der Veranstaltung die Grundlagen der rechnergestützten Methoden der geometrischen Modellierung und Simulation. Sie verstehen verschiedene parametrische Kurven- und Oberflächenrepräsentationen und können diese auswerten und miteinander vergleichen. Weiter kennen Sie klassische Datenstrukturen und Algorithmen aus dem Computer Aided Design (CAD). Sie sind in der Lage, diese Techniken praktisch umzusetzen und damit 3D-Geometrie im Rechner darzustellen und zu visualisieren.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Grundwissen in Informatik</p>				
5	<b>Prüfungsform</b> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"><li>• [20-00-0140-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0140-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Vorlesungsfolien</p> <p>Lee: Principles of CAD / CAM / CAE Systems, Addison-Wesley.</p> <p>Piegl, Tiller: The NURBS Book, Springer Verlag.</p> <p>Farin: Kurven und Flächen im Computer Aided Geometric Design, vieweg</p> <p>Shah, Mäntylä: Parametric and Feature-based CAD/CAM, Wiley &amp; Sons</p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Programmierung Massiv-Paralleler Prozessoren					
<b>Modul Nr.</b> 20-00-0419	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0419-iv	Programmierung Massiv-Paralleler Prozessoren	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> - Grundlagen massiv-paralleler Hardware mit einem Schwerpunkt auf modernen Beschleunigern - parallele Algorithmen - effiziente Programmierung massiv-paralleler Systeme - praktische Programmierprojekte mit Co-Betreuung durch einen Wissenschaftler aus seiner Anwendungsdomain				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung sind Studierende dazu in der Lage, Problemstellungen im Kontext massiv-paralleler Systeme zu analysieren. Sie können selbständig neue Anwendungen entwickeln und ihre Performanz systematisch verbessern. Sie verstehen grundlegende parallele Algorithmen und Programmierparadigmen und können sich selbständig aktuelle Literatur erarbeiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  solide Programmierkenntnisse in C/C++  Kenntnisse in paralleler Programmierung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0419-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0419-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Physikalisch-basierte Animation					
<b>Modul Nr.</b> 20-00-0682	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0682-iv	Physikalisch-basierte Animation	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> 1. Grundlagen der physikalisch-basierten Animation <ul style="list-style-type: none"><li>- Anwendungen</li><li>- Simulationsmodelle</li><li>- Definition holonomer und nichtholonmer Zwangsbedingungen</li><li>- Bewegungsgleichungen für Partikel</li><li>- Gewöhnliche Differentialgleichungen</li><li>- Numerische Integrationsverfahren</li></ul> 2. Partikelsysteme <ul style="list-style-type: none"><li>- Aufbau von Partikelsystemen</li><li>- Simulation physikalischer Effekte</li></ul> 3. Simulation von Haaren <ul style="list-style-type: none"><li>- Haarmodelle</li><li>- Simulationsverfahren</li><li>- Haar-Haar Interaktion</li></ul> 4. Simulation von Kleidung <ul style="list-style-type: none"><li>- Masse-Feder-Systeme</li><li>- Finite-Elemente-Methoden</li><li>- Positionsbasierte Verfahren</li></ul> 5. Simulation von Weichkörpern <ul style="list-style-type: none"><li>- Generierung von Volumennetzen</li><li>- Masse-Feder-Systeme</li><li>- Finite-Elemente-Methoden</li><li>- Positionsbasierte Verfahren</li><li>- Volumenerhaltung</li></ul> 6. Starrkörper <ul style="list-style-type: none"><li>- Grundlagen</li><li>- Bewegungsgleichungen für Starrkörper</li></ul>				

	<ul style="list-style-type: none"> <li>- Simulation von Gelenken</li> </ul> <p>7. Kollisionserkennung</p> <ul style="list-style-type: none"> <li>- Hüllkörper</li> <li>- Hüllkörperhierarchien</li> <li>- Zellrasterverfahren</li> <li>- Kollisionstests für Starrkörper</li> <li>- Kollisionstests für deformierbare Körper</li> <li>- Kontinuierliche Kollisionserkennung</li> <li>- Bildbasierte Verfahren</li> </ul> <p>8. Brüche</p> <ul style="list-style-type: none"> <li>- Animation von Brüchen mit Bruchmustern</li> <li>- Simulation spröder Brüche</li> <li>- Anpassung des Simulationsnetzes</li> </ul>
3	<p><b>Qualifikationsziele / Lernergebnisse</b></p> <p>Studierende kennen nach einem erfolgreichen Besuch der Veranstaltung Mehrkörpersysteme und diskrete und kontinuierliche deformierbare Simulationsmodelle. Sie verstehen die numerischen Simulationsverfahren sowie deren jeweiligen Anwendungsbereiche und können diese Verfahren anwenden. Sie haben einen grundlegenden Überblick über Verfahren der Echtzeitsimulation in der Computergraphik.</p>
4	<p><b>Voraussetzung für die Teilnahme</b></p> <p>Empfohlen: Grundlegende Kenntnisse von Numerik, Algorithmen und Datenstrukturen, Computergraphik</p>
5	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0682-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0682-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p>



	B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> wird in der Vorlesung bekannt gegeben
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Verteilte Geometrieverarbeitung					
<b>Modul Nr.</b> 20-00-1075	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1075-iv	Verteilte Geometrieverarbeitung	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> * Grundlagen und Algorithmen der Geometrieverarbeitung: Smoothing, Remeshing, Delaunay-Triangulierung, Parametrisierung, Texturierung, u.a. * Einführung in Big Data und Cloud Computing * Indexstrukturen für den schnellen Zugriff auf massive Geometriedatenmengen: Quad tree, R-tree, Space-filling curves, u.a. * Verteilte und cloud-basierte Datenspeicherung * Architekturen für verteilte Verarbeitungspipelines * Programmiermodelle für verteilte Algorithmen (z.B. MapReduce) * Technologien und Frameworks für die verteilte Datenverarbeitung (z.B. Spark, Vert.x) und Geometrieverarbeitung (Draco, u.a.) * Deployment von verteilten Anwendungen in die Cloud * Ergänzend gibt es praktische und theoretische Übungen				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung besitzen die Studierenden grundlegende Kenntnisse zur Geometrieverarbeitung sowie zur verteilten, cloud-basierten Verarbeitung sehr großer Datenmengen im Allgemeinen. Sie sind in der Lage, selbstständig skalierbare Anwendungen zu entwickeln und diese in der Cloud auszuführen, um die Geometrieverarbeitung zu parallelisieren und damit die Performance zu erhöhen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  * Programmierkenntnisse in Java oder anderen JVM-Sprachen * Grundlegende Algorithmen und Datenstrukturen				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1075-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1075-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p>
<b>10</b>	<p><b>Kommentar</b></p>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Visual Computing**

**Wahlbereich Computer Vision und Machine  
Learning**

## Modulbeschreibung

<b>Modulname</b> Data Mining und Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0052	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0052-iv	Data Mining und Maschinelles Lernen	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b>				
	Durch die rasante Entwicklung der Informationstechnologie sind immer größere Datenmengen verfügbar. Diese enthalten oft implizites Wissen, das, wenn es bekannt wäre, große wirtschaftliche oder wissenschaftliche Bedeutung hätte. Data Mining ist ein Forschungsgebiet, das sich mit der Suche nach potentiell nützlichem Wissen in großen Datenmengen beschäftigt, und Maschinelles Lernverfahren gehören zu den Schlüsseltechnologien innerhalb dieses Gebiets.				
	Die Vorlesung bietet eine Einführung in das Gebiet des Maschinellen Lernens unter dem besonderen Aspekt des Data Minings. Es werden Verfahren aus verschiedenen Paradigmen des Maschinellen Lernens mit exemplarischen Anwendungen vorgestellt. Um das Wissen zu operationalisieren, werden in den Übungen prak-tisch-e Erfahrungen mit Lernalgorithmen gesammelt.				
	• Einführung (Grundbegriffe, Lernprobleme, Konzepte, Beispiele, Repräsentation)				
	• Regel-Lernen				
	◦ Lernen einzelner Regeln (Generalisierung und Spezialisierung, Strukturierte Hypothesenräume, Version Spaces)				
	◦ Lernen von Regel-Mengen (Covering Strategie, Evaluierungsmaße für Regeln, Pruning, Mehr-Klassenprobleme)				
	• Evaluierung und kosten-sensitives Lernen (Accuracy,X-Val,ROC-Kurven,Cost-Sensitive Learning)				
	• Instanzenbasiertes Lernen (kNN,IBL,NEAR,RISE)				
	• Entscheidungsbaum-Lernen (ID3, C4.5, etc.)				
	• Ensemble-Methoden (Bias/Variance, Bagging, Randomization, Boosting, Stacking, ECOCs)				
	• Pre-Processing (Feature Subset Selection, Diskretisierung, Sampling, Data Cleaning)				
• Clustering und Lernen von Assoziationsregeln (Apriori)					

3	<p><b>Qualifikationsziele / Lernergebnisse</b></p> <p>Nach der erfolgreichen Absolvierung dieser Lehrveranstaltung sind die Studierenden in der Lage</p> <ul style="list-style-type: none"> <li>• grundlegende Techniken des Data Mining und Maschinellen Lernens zu verstehen und erklären</li> <li>• praktische Data Mining Systeme selbständig einsetzen und deren Stärken und Schwächen verstehen</li> <li>• neue Entwicklungen auf diesem Gebiet kritisch beurteilen</li> </ul>
4	<p><b>Voraussetzung für die Teilnahme</b></p>
5	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0052-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0052-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning  M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>

9	<b>Literatur</b> <ul style="list-style-type: none"> <li>• Mitchell: Machine Learning, McGraw-Hill, 1997</li> <li>• Ian H. Witten and Eibe Frank: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan-Kaufmann, 1999</li> </ul>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Bildverarbeitung					
<b>Modul Nr.</b> 20-00-0155	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0155-iv	Bildverarbeitung	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> Überblick über die Grundlagen der Bildverarbeitung: - Bildeigenschaften - Bildtransformationen - einfache und komplexere Filterung - Bildkompression, - Segmentierung - Klassifikation				
3	<b>Qualifikationsziele / Lernergebnisse</b> Noch erfolgreichem Besuch der Veranstaltung haben die Studierenden einen Überblick über die Funktionsweise und die Möglichkeiten der modernen Bildverarbeitung. Studierende sind dazu in der Lage, einfache bis mittlere Bildverarbeitungsaufgaben selbständig zu lösen.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0155-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				



<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0155-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> <ul style="list-style-type: none"> <li>• Gonzalez, R.C., Woods, R.E., "Digital Image Processing", Addison- Wesley Publishing Company, 1992</li> <li>• Haberaecker, P., "Praxis der Digitalen Bildverarbeitung und Mustererkennung", Carl Hanser Verlag, 1995</li> <li>• Jaehne, B., "Digitale Bildverarbeitung", Springer Verlag, 1997</li> </ul>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Computer Vision					
<b>Modul Nr.</b> 20-00-0157	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0157-iv	Computer Vision	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Grundlagen der Bildformierung</li><li>• Lineare und (einfache) nichtlineare Bildfilterung</li><li>• Grundlagen der Mehransichten-Geometrie</li><li>• Kamerakalibrierung &amp; -posenschätzung</li><li>• Grundlagen der 3D-Rekonstruktion</li><li>• Grundlagen der Bewegungsschätzung aus Videos</li><li>• Template- und Unterraum-Ansätze zur Objekterkennung</li><li>• Objektklassifikation mit Bag of Words</li><li>• Objektdetektion</li><li>• Grundlagen der Bildsegmentierung</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Studierende beherrschen nach erfolgreichem Besuch der Veranstaltung die Grundlagen der Computer Vision. Sie verstehen grundlegende Techniken der Bild- und Videoanalyse, und können deren Annahmen und mathematische Formulierungen benennen, sowie die sich ergebenden Algorithmen beschreiben. Sie sind in der Lage diese Techniken praktisch so umzusetzen, dass sie grundlegende Bildanalyseaufgaben an Hand realistischer Bilddaten lösen können.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Der vorherige Besuch von „Visual Computing“ oder einer vergleichbaren Veranstaltung</p>				
5	<b>Prüfungsform</b> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"><li>• [20-00-0157-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0157-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning  M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden regelmässig aktualisiert und beinhalten beispielsweise:</p> <ul style="list-style-type: none"> <li>• R. Szeliski, "Computer Vision: Algorithms and Applications", Springer 2011</li> <li>• D. Forsyth, J. Ponce, "Computer Vision -- A Modern Approach", Prentice Hall, 2002</li> </ul>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Statistisches Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0358	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0358-iv	Statistisches Maschinelles Lernen	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> - Statistische Methodik für das Maschinelle Lernen - Auffrischung zu Statistik, Optimierung und Linearer Algebra - Bayes'sche Entscheidungstheorie - Wahrscheinlichkeitsdichtenschätzung - Nichtparametrische Modelle - Mixtur Modelle und der EM-Algorithmus - Lineare Modelle zur Klassifikation und Regression - Statistische Lerntheorie - Kernel Methoden zur Klassifikation und Regression				
3	<b>Qualifikationsziele / Lernergebnisse</b> Die Lehrveranstaltung ist eine systematische Einführung in die Grundlagen und Methodik des statistischen maschinellen Lernens. Nach erfolgreichen Abschluss der Lehrveranstaltung, verstehen Studierende die wichtigsten Methoden und Ansätze des Statistischen Maschinellen Lernens. Sie können maschinelle Lernverfahren anwenden, um eine Vielzahl neuer Probleme zu lösen.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0358-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0358-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> <ol style="list-style-type: none"> <li>1. C.M. Bishop, Pattern Recognition and Machine Learning (2006), Springer</li> <li>2. K.P. Murphy, Machine Learning: a Probabilistic Perspective (expected 2012), MIT Press</li> <li>3. D. Barber, Bayesian Reasoning and Machine Learning (2012), Cambridge University Press</li> <li>4. T. Hastie, R. Tibshirani, and J. Friedman (2003), The Elements of Statistical Learning, Springer Verlag</li> <li>5. D. MacKay, Information Theory, Inference, and Learning Algorithms (2003), Cambridge University Press</li> <li>6. R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification (2nd ed. 2001), Wiley-Interscience</li> <li>7. T.M. Mitchell, Machine Learning (1997), McGraw-Hill</li> </ol>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Medizinische Bildverarbeitung					
<b>Modul Nr.</b> 20-00-0379	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0379-v1	Medizinische Bildverarbeitung	3	integrierte Veranstaltung	2
2	<b>Lerninhalt</b> Die Vorlesung gliedert sich in zwei Teile. In der ersten Hälfte der Vorlesung wird die Funktionsweise von Geräten, welche medizinische Bilder liefern (CT, MRI, PET, SPECT, Ultraschall), erklärt. In der zweiten Hälfte werden verschiedene Bildverarbeitungsmethoden erklärt, welche typischerweise für die Bearbeitung medizinischer Bilder eingesetzt werden.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Noch erfolgreichem Besuch der Veranstaltung haben die Studierenden einen Überblick über die Funktionsweise und die Möglichkeiten der modernen medizinischen Bildverarbeitung. Studierende sind dazu in der Lage, einfache bis mittlere medizinische Bildverarbeitungsaufgaben selbständig zu lösen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Mathematische Grundlagen sind dringend empfehlenswert. Ferner wird empfohlen, die Vorlesung „Bildverarbeitung“ oder eine vergleichbare Veranstaltung vorher besucht zu haben.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0379-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				

	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0379-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> 1) Heinz Handels: Medizinische Bildverarbeitung 2) 2) Gonzalez/Woods: Digital Image Processing (last edition) 3) 3) Bernd Jähne: Digitale Bildverarbeitung. 6. überarbeitete und erweiterte Auflage. Springer, Berlin u. a. 2005, ISBN 3-540-24999-0. 4) Kristian Bredies, Dirk Lorenz: Mathematische Bildverarbeitung. Einführung in Grundlagen und moderne Theorie. Vieweg+Teubner, Wiesbaden 2011, ISBN 978-3-8348-1037-3.
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Computer Vision II					
<b>Modul Nr.</b> 20-00-0401	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0401-iv	Computer Vision II	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Computer Vision als (probabilistische) Inferenz</li><li>• Robuste Schätzung und Modellierung</li><li>• Grundlagen der Bayes'schen Netze und Markov'schen Zufallsfelder</li><li>• Grundlegende Inferenz- und Lernverfahren der Computer Vision</li><li>• Bildrestaurierung</li><li>• Stereo</li><li>• Optischer Fluß</li><li>• Bayes'sches Tracking von (artikulierten) Objekten</li><li>• Semantische Segmentierung</li><li>• Aktuelle Themen der Forschung</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Studierende haben nach erfolgreichem Besuch der Veranstaltung ein vertieftes Verständnis der Computer Vision. Sie formulieren Fragestellungen der Bild- und Videoanalyse als Inferenzprobleme und berücksichtigen dabei Herausforderungen reeller Anwendungen, z.B. im Sinne der Robustheit. Sie lösen das Inferenzproblem mittels diskreter oder kontinuierlicher Inferenzalgorithmen, und wenden diese auf realistische Bilddaten an. Sie evaluieren die anwendungsspezifischen Ergebnisse quantitativ.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Der vorherige Besuch von „Visual Computing“ und „Computer Vision I“ oder vergleichbaren Veranstaltungen ist empfohlen.</p>				
5	<b>Prüfungsform</b> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"><li>• [20-00-0401-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				



	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0401-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning  M. Sc. IT Sicherheit</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden regelmässig aktualisiert und beinhalten beispielsweise:</p> <ul style="list-style-type: none"> <li>• S. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012</li> <li>• R. Szeliski, "Computer Vision: Algorithms and Applications", Springer 2011</li> </ul>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b>					
Statistical Relational Artificial Intelligence: Logic, Probability, and Computation					
<b>Modul Nr.</b> 20-00-1011	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1011-iv	Statistical Relational Artificial Intelligence: Logic, Probability, and Computation	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> + logische Programmierung + Lernen von logischen Programmen aus Daten + Probabilistische Graphische Modelle: Inferenz und Lernen + Statistisch-Relationale Modelle wie z.B. ProbLog und Markov Logic Networks + Schlussfolgern in statistisch-relationalen Modellen + Lernen von statistisch-relationalen Modellen aus Daten + Relationale lineare und quadratische Programme				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Lehrveranstaltung ist eine systematische Einführung in die Grundlagen und Methodik des statistisch-relationalen Lernens und Künstlichen Intelligenz: Das Studium und Design von intelligenten Agenten, die in verrauschten Welten agieren, die aus Individuen (Objekte, Dinge) und komplexe Beziehungen zwischen den Individuen bestehen. Nach erfolgreichem Abschluss der Lehrveranstaltung verstehen Studierende die wichtigsten Methoden und Ansätze in der statistisch-relationalen Künstlichen Intelligenz. Sie verstehen die grundlegenden Herausforderungen von relationalen Domänen. Sie kennen aktuelle Ansätze, um diese Herausforderungen zu lösen. Sie sind außerdem in der Lage ihre Kenntnisse auf aktuelle Probleme anzuwenden.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Der vorherige Besuch von “Statistisches Maschinelles Lernen” und “Probabilistische Graphische Modelle” oder vergleichbarer Veranstaltungen ist empfohlen, ist aber keine Voraussetzung.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1011-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul>				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>
<b>6</b>	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
<b>7</b>	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1011-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik  M. Sc. Informatik  M. Sc. Computer Science  M. Sc. Autonome Systeme und Robotik  M. Sc. Artificial Intelligence and Machine Learning</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p> <p>Literaturempfehlungen werden regelmäßig aktualisiert und beinhalten beispielsweise:</p> <p>Luc De Raedt, Kristian Kersting, Sriraam Natarajan, David Poole (2016): Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan &amp; Claypool Publishers, ISBN: 9781627058414.</p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

Modulname					
Deep Learning für medizinische Bildgebung					
Modul Nr. 20-00-1014	Leistungspunkte 5 CP	Arbeitsaufwand 150 h	Selbststudium 105 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1014-iv	Deep Learning für medizinische Bildgebung	5	Integrierte Veranstaltung	3
2	Lerninhalt Formulierung der medizinischen Bildsegmentierung, Computergestützte Diagnostik und chirurgische Planung als Probleme des maschinellen Lernens, Deep Learning für medizinische Bildsegmentierung, Deep Learning für computergestützte Diagnostik, Chirurgische Planung von präoperativen Bildern mit Deep Learning, Tool-Präsenz Erkennung und Lokalisierung von endoskopischen Videos durch Deep Learning, Adversarial Beispiele für medizinische Bildgebung, Generative Adversarial Networks für Medizinische Bildgebung.				
3	Qualifikationsziele / Lernergebnisse Nach erfolgreichem Abschluss des Kurses sind die Studierenden in der Lage sein, alle Komponenten der Formulierung eines medizinischen Bildanalyseproblems als Problem des Maschinellen Lernens zu verstehen. Sie sind auch in der Lage sein, fundierte Entscheidungen über die Wahl eines universellen Deep Learning Paradigmas für ein gegebenes medizinische Bildanalyseproblem zu treffen.				
4	Voraussetzung für die Teilnahme Empfohlen:  - Programmierkenntnisse - Verständnis des algorithmischen Designs - Kenntnisse aus dem Bereich Lineare Algebra - Der vorherige Besuche von „Bildverarbeitung“, „Computer Vision I“ und „Statistisches Maschinelles Lernen“ oder vergleichbarer Veranstaltungen				
5	Prüfungsform Bausteinbegleitende Prüfung:  • [20-00-1014-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1014-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Deep Learning: Architectures & Methods					
Modul Nr. 20-00-1034	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1034-iv	Deep Learning: Architectures & Methods	6	Integrierte Veranstaltung	4
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Auffrischung des Hintergrundwissens</li><li>• Deep Feedforward Netze</li><li>• Regularisierung im Deep Learning</li><li>• Optimierung zum Training tiefer Netze</li><li>• Convolutional tiefe Netze</li><li>• Modellierung von Sequenzen durch Rekordernte und Rekursive Netze</li><li>• Lineare Faktor Modelle</li><li>• Autoenkoder</li><li>• Repräsentationslernen</li><li>• Strukturierte Probabilistische Modelle zum Deep Learning</li><li>• Monte Carlo Methoden</li><li>• Approximative Inferenz</li><li>• Tiefe generative Modelle</li><li>• Deep Reinforcement Learning</li><li>• Deep Learning in Vision</li><li>• Deep Learning in NLP</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Dieser Kurs richtet sich an Studierende mit fortgeschrittenem Erfahrung im maschinellen Lernen und vermittelt diesen Studierenden das notwendige Wissen, um eigenständig Forschungsprojekte im Bereich der Deep Learning durchzuführen, z.B. im Rahmen einer Bachelor- oder Masterarbeit. Dies betrifft sowohl ein grundlegendes Verständnis der algorithmischen Ansätze zum Deep Learning als auch die der Architekturen der tiefen tiefen Netze.</p>				
4	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Der vorherige Besuch von „Statistisches Maschinelles Lernen“ und „Data Mining und Maschinelles Lernen“ oder vergleichbarer Veranstaltungen</p>				

<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1034-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1034-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Reinforcement Learning: Von Grundlagen zu den tiefen Ansätzen					
<b>Modul Nr.</b> 20-00-1047	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1047-iv	Reinforcement Learning: Von Grundlagen zu den tiefen Ansätzen	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Auffrischung des Hintergrundwissens</li><li>• Black box Reinforcement Learning</li><li>• Modellierung als Bandit, Markov Decision Processes und Partially Observable Markov Decision Processes</li><li>• Optimale Steuerung und Regelung</li><li>• Modellernen</li><li>• Wertefunktionslernen</li><li>• Policy Search</li><li>• Tiefe Wertefunktion Methoden</li><li>• Tiefe Policy Search Methoden</li><li>• Exploration vs Exploitation</li><li>• Hierarchisches Reinforcement Learning</li><li>• Intrinsische Motivation</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Dieser Kurs richtet sich an Studierende mit erster Erfahrung im maschinellen Lernen und vermittelt diesen Studierenden das notwendige Wissen, um eigenständig Forschungsprojekte im Bereich der Reinforcement Learning durchzuführen, z.B. im Rahmen einer Bachelor- oder Masterarbeit. Dies betrifft sowohl ein grundlegendes Verständnis der algorithmischen Ansätze zum Reinforcement Learning als auch Anwendungen von tiefen Netzen.</p>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen:</p> <p>Gute Programmierkenntnisse in Python. Der vorherige Besuch von „Statistisches Maschinelles Lernen“ oder einer vergleichbaren Veranstaltung ist hilfreich aber nicht zwingend erforderlich</p>				



<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1047-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-1047-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Mensch- und Identitätsfokussiertes Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-1118	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1118-iv	Mensch- und Identitätsfokussiertes Maschinelles Lernen	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b>				
	Hintergründe und Konzepte von Human-Centric Machine Learning: Das Ziel von Identität und Human-Centric Machine Learning. Die Unterschiede zwischen Identitätslernen und anderen gängigen Klassifikationsarten.				
	Repräsentationsextraktion für subjektbezogene Daten: Methoden für die Feature Erstellung für identitätsbezogene Anwendungen. Grundlagen und Hintergründe für handgefertigte oder Deep Learning Features.				
	Deep-Learning Strategien für Identitätsrepräsentationen: Erlernen von Identitätsrepräsentationen mit Hilfe von Deep Learning. Lernstrategien und Loss-Funktionen.				
	Netzwerkarchitekturen und identitätsspezifische Komponenten.				
	Knowledge Transfer und Distillation: Transfer Learning und Identitätsrepräsentation. Konzepte und Anwendungen von Knowledge Distillation.				
	Effizientes Machine Learning: Beziehung zwischen Ressourcenbeschränkungen, Green-AI und Deep Learning. Methoden zum Aufbau effizienter Lösungen für Maschinelles Lernen.				
	Synthetische Identität: Die Notwendigkeit einer synthetischen Identität. Synthetische Identität als Adversarial. Generierung synthetischer identitätsgesteuerter Daten unter verschiedenen Einschränkungen.				
	Machine Learning Biases: Analyse der demografischen Fairness und der Ursachen der Fairnessprobleme. ML-basierte Abmilderung von demografischen Bias.				
	Privatsphäre erlernen: Analyse von unbeabsichtigt gelernten Informationen. Lernstrategien zur gezielten Unterdrückung von Informationen auf verschiedenen Repräsentationsebenen.				

	<p>Data Utility: Verständnis der Auswirkungen von Data Utility im Lernprozess. Verstehen von Sample Utility im Betrieb. ML-Konzepte und Strategien zur Schätzung von Sample Utilities.</p> <p>Angriffe auf Sample-Level: Überblick über Adversarial, Sample Manipulation und andere Angriffe auf Human-Centric ML. Deep Learning Konzepte, Netzwerklöcke und LossStrategien um Sample-Level Angriffe zu erkennen und zu umgehen.</p> <p>Explainability: Überblick über den Bedarf von Explainability in verschiedenen Entscheidungsprozessen. Verschiedene Strategien um Explainability für Themen aus vergangenen Vorlesungen.</p>
<b>3</b>	<p><b>Qualifikationsziele / Lernergebnisse</b></p> <p>Nach erfolgreichem Besuch des Kurses sind die Studierenden mit Konzepten des maschinellen Lernens im Umgang mit personen- und identitätsbezogenen Informationen vertraut. Sie verstehen die grundlegenden Techniken für die Extraktion subjektspezifischer Repräsentationen, einschließlich der damit verbundenen Konzepte für Knowledge Transfer und Distillation. Die Studierenden haben ein Verständnis für demografisch bedingte Verzerrungen beim maschinellen Lernen und Datenschutzbedenken zu Function-Creep erlangt, einschließlich der wichtigsten Konzepte zur Abschwächung dieser Probleme. Sie kennen die Anforderungen und Techniken, die für ein eingebettetes und effizientes HumanCentric Machine Learning erforderlich sind. Ebenfalls sind sie mit den Auswirkungen von Data Utility im Lernprozess und dem Hauptkonzept zur Schätzung der Utility von subjektbezogenen Daten vertraut. Sie werden fundiertes Wissen über die Erklärungsmethoden für ML-Entscheidungen auf der Grundlage von identitätsbezogenen Daten erlangen. Die Studierenden werden in die Konzepte der KI-Ethik und der KIRegulierung im Zusammenhang mit der Verarbeitung und Speicherung personenbezogener Daten eingeführt. Sie sind in der Lage, diese Techniken zur Lösung grundlegender Aufgaben im Bereich von Identitäts- und Human-Centric Machine Learning auf realistische Probleme anzuwenden.</p>
<b>4</b>	<p><b>Voraussetzung für die Teilnahme</b></p> <p>Empfohlen: Der vorherige Besuch der Veranstaltung „Visual Computing“ oder einer vergleichbaren Veranstaltung. Grundlagen in Mathematik und Wahrscheinlichkeitsrechnung.</p>
<b>5</b>	<p><b>Prüfungsform</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-1118-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)</p>

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1118-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Affective Computing					
<b>Modul Nr.</b> 20-00-1120	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1120-iv	Affective Computing	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>- Einführung in das Affective Computing mit einem Überblick über die Anwendung in Unterhaltung, Gesundheit und Pädagogik</li><li>- Emotionstheorien: Psychologie, Kognitionswissenschaft und Neurowissenschaft</li><li>- Diskussion über Möglichkeiten, wie Maschinen Emotionen "haben" können</li><li>- Experimenteller Aufbau, Methodik und Analyse</li><li>- Emotionen und das Gehirn</li><li>- Körperlicher Ausdruck von Emotionen</li><li>- Synthese von emotionalem Verhalten</li><li>- Emotionen und soziale Interaktion</li><li>- Persönlichkeit und Kulturen</li><li>- Emotionserkennung in Text, Sprache und Gesicht</li><li>- Praktische Programmiererfahrung für Affective Computing</li><li>- Vorurteile und Ethik des Affective Computing</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Dieser Kurs zielt darauf ab, die Theorien, Methoden und Anwendungen rund um Affective Computing in einer interdisziplinären Perspektive zu lehren. Nach erfolgreichem Abschluss des Kurses verstehen die Studierenden affektive Interaktionen und deren Auswirkungen auf die Mensch-Computer-Interaktion, lernen Methoden zur Erhebung, Analyse und Auswertung affektiver Verhaltensdaten anzuwenden. Sie demonstrieren Kenntnisse zur computergestützten Analyse, Synthese und Erkennung menschlicher affektiver Verhaltensdaten und zum Entwurf emotionssensibler interaktiver Technologien wie Interaktionen mit virtuellen Agenten, Robotern und Spielen. Sie gewinnen praktische Erfahrung mit den Rahmenbedingungen für menschliches Affekt- und Verhaltensverständnis und ein Bewusstsein für potenzielle Verzerrungen in Daten sowie mögliche Gefahren im Umgang mit sensiblen personenbezogenen Daten.				

<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen werden: <ul style="list-style-type: none"> <li>- Programmierkenntnisse</li> <li>- Statistisches Maschinelles Lernen oder Einführung in die Künstliche Intelligenz</li> </ul>
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1120-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1120-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Visual Computing**

**Wahlbereich Integrated Methods of Graphics  
and Vision**

## Modulbeschreibung

<b>Modulname</b> Virtuelle und Erweiterte Realität					
<b>Modul Nr.</b> 20-00-0160	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0160-iv	Virtuelle und Erweiterte Realität	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Im Rahmen dieser Lehrveranstaltung werden zuerst die Grundlagen, Begriffsbildungen und Referenzmodelle zur Einordnung der Thematik im Rahmen der Computer-Graphik/Computer-Vision aufgezeigt. Aufbauend darauf werden die besonderen Technologien, Algorithmen und Standards der Augmented Reality (AR) und der Virtual Reality (VR) behandelt. Dazu gehören: <ul style="list-style-type: none"><li>• Datenschnittstellen (Standards, Vorverarbeitung, Systeme, etc.)</li><li>• Interaktionstechniken (z.B. Interaktion mit Hilfe von Rangekameras)</li><li>• Darstellungsverfahren (z.B. Echtzeit-Rendering)</li><li>• Web-basierte VR/AR</li><li>• Computer-Vision-basiertes Tracking für Augmented-Reality</li><li>• Augmented Reality mit Rangekamera-Technologien</li><li>• Augmented Reality auf Smartphonesystemen</li></ul> Schließlich werden diese Techniken an Beispielen aktueller Forschungsarbeiten aus den Bereichen „AR/VR-Wartungsunterstützung“ und „AR/VR-gestützte Präsentation von Kulturgütern“ dokumentiert.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung die Anforderungen und Problematiken von Virtual/Augmented Reality und sie wissen, für welche Problemstellungen diese Technologien eingesetzt werden können. Sie kennen die Standards, mit deren Hilfe VR/AR-Anwendungen spezifiziert werden, insb. wissen die Studierenden, welche Computer-Vision-Technologien eingesetzt werden können, um in verschiedenen Umgebungen die Kamerapose stabil zu tracken.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen der Graphischen Datenverarbeitung (GDV)				



<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0160-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0160-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Dörner, R., Broll, W., Grimm, P., Jung, B. Virtual und Augmented Reality (VR / AR)
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Informationsvisualisierung und Visual Analytics					
<b>Modul Nr.</b> 20-00-0294	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0294-iv	Informationsvisualisierung und Visual Analytics	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Diese Vorlesung wird eine detaillierte Einführung in die Informationsvisualisierung geben, um sich dann intensiv den wissenschaftlichen Fragestellungen und praxisnahen Anwendungsszenarien von Visual Analytics zu widmen. •Überblick der Informationsvisualisierung und Visual Analytics (Definitionen, Modelle, Historie) •Datenpräsentierung und Datentransformation •Abbildung von Daten auf visuelle Strukturen • Visuelle Repräsentierungen und Interaktion fuer bivariate, multivariate Daten, Zeitreihen, Graphen und Geographische Daten • Grundlagen von Data Mining •Grundlagen von Visual Analytics: - Analytische Beweisführung - Data Mining • Evaluation von Visual Analytics Systemen  Anwendungsgebiete: Medizin, Biologie, Finanzen und Wirtschaft, Meteorologie, Rettungsdienst,....				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende können nach erfolgreichem Besuch der Veranstaltung  •Informationsvisualisierungsmethoden für verschiedene Datentypen benutzen  •interactive Visualisierungssysteme für Daten aus verschiedenen Anwendungsgebieten designen  •Visualisierung und automatische Datenverarbeitung kombinieren um Big Data Probleme zu lösen				

	<ul style="list-style-type: none"> <li>• Wissen über Hauptcharakteristika menschlicher visuellen Wahrnehmung in Informationsvisualisierung und Visual Analytics anwenden</li> <li>• geeignete Evaluationsmethode für spezifische Situationen und Szenarien auswählen</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Interesse an Methoden der Computergrafik und Visualisierung
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0294-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0294-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M.Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird in der Vorlesung bekanntgegeben. Beispiele für verwendete Literatur könnten sein: C. Ware: Information Visualization: Perception for Design Ellis et al: Mastering the Information Age
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Ambient Intelligence					
<b>Modul Nr.</b> 20-00-0390	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0390-iv	Ambient Intelligence	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Die Vorlesung führt in aktuelle Entwicklungen von Ambient Intelligence ein. Im Vordergrund der Vorlesung steht die Mensch-Maschine-Interaktion (MMI) in intelligenten Umgebungen in einem allgegenwärtigen Informationsraum, wie sie beispielsweise zunehmend durch eingebettete Systeme in alltägliche Gebrauchsobjekte gegeben ist. Spezieller Fokus wird auf den mobilen Aspekt eines allgegenwärtigen Informationszugriffs und der Informationsaufbereitung und -darstellung in mobilen Endgeräten gelegt. Dabei soll einerseits ein Einblick in die grundlegenden Technologien, Anwendungen und Experimente gegeben werden und andererseits (nicht im Schwerpunkt) auch die sozio-kulturellen Implikationen und Aspekte neuer Ambient Intelligence Lösungen diskutiert werden. Zusätzliche Themen der Vorlesung sind System-Architekturen für verteilte Umgebungen, Kontext-Awareness und Kontext-Management, Benutzermodelle und deren Implikationen, Sensornetzwerke und Interaktionstechniken. Die Vorlesung wird Beispiele aktueller Projekte diskutieren und die internationalen Forschungslinien von Ambient Intelligence beleuchten.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nachdem Studierende die Veranstaltung erfolgreich besucht haben, können sie Technologietrends und Forschungserkenntnisse im Bereich Ambient Intelligence beschreiben. Die wichtigsten Konzepte zur Realisierung „intelligenter Umgebungen“ - intelligente Netzwerke und Objekte, Techniken der erweiterten, mobilen Realität, ubiquitäre und allgegenwärtige Informationsräume, nomadische Kommunikationen, Echt-Zeit-Kommunikation und relevante Middleware, Eingebettete Systeme, Sensor Netzwerke und Wearable Computing - können diskutiert und eingeordnet werden. Nach Abschluss der zugehörigen Übung können Studierende die Projektphasen der Entwicklung einer Ambient-Intelligence Anwendung eigenständig planen und realisieren.				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch von “Visual Computing“ und „Multimodale Interaktion mit intelligenten Umgebungen“ oder vergleichbarer Veranstaltungen
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0390-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0390-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird jeweils passend zu den aktuellen Themen bekanntgegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Medizinische Visualisierung					
<b>Modul Nr.</b> 20-00-0467	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0467-iv	Medizinische Visualisierung	6	integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Medizinische Bilddaten; Bildaufbereitung; Medizinische Visualisierung mit VTK; Indirekte Volumenvisualisierung; Direkte Volumenvisualisierung; Transfer-Funktionen; Interaktive Volumenvisualisierung; Illustratives Rendering; Beispiel: Visualisierung von Tensor-Bilddaten; Beispiel: Visualisierung von Baumstrukturen; Beispiel: Virtuelle Endoskopie; Beispiel: Bildgestützte Chirurgie				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung Techniken der Volumenvisualisierung. Sie verstehen die Notwendigkeit der Bildverbesserung für die Visualisierung. Sie können das "Visualization Toolkit" (VTK) anwenden, um mit dessen Hilfe Anwendungen für die Visualisierung von medizinischen Bilddaten für Diagnose, Planung und Therapie zu erstellen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: GDV I, (Medizinische) Bildverarbeitung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0467-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				

	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0467-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Preim, Botha: Visual Computing for Medicine
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname Capturing Reality					
Modul Nr. 20-00-0489	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0489-iv	Capturing Reality	6	integrierte Veranstaltung	4
2	<b>Lerninhalt</b> Dieser Kurs deckt ein breites Spektrum von Techniken zur Digitalisierung und Modellierung unserer Welt mit einem Fokus auf Anwendungen in der Computergraphik und Computer Vision ab. Dies beinhaltet insbesondere: - grundlegende Werkzeuge und Kalibrationstechniken für die Digitalisierung - Digitalisierungs- und Modellierungstechniken für verschiedenste Objekt- und Szeneneigenschaften (z.B. Geometrie, Reflexionseigenschaften) - grundlegende mathematische Modellierungs- und Optimierungstechniken - Implementierung und praktische Anwendung einer Reihe von Techniken				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Besuch der Veranstaltung sind Studierende dazu in der Lage, Digitalisierungs- und Modellierungsprobleme für Objekte und Szenen in Computergraphik und Computer Vision sowie die zugrunde liegenden Techniken zu analysieren. Sie können selbständig neue Versuchsaufbauten entwickeln, Experimente durchführen und die Ergebnisse auswerten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Der vorherige Besuch der Veranstaltungen „Graphische Datenverarbeitung I“ oder „Computer Vision I“ oder vergleichbaren Veranstaltungen sowie grundlegende Programmierkenntnisse in C/C++				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0489-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				



	Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0489-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul> <p>In dieser Veranstaltung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25(2) der 6. Novelle der Allgemeinen Prüfungsbestimmungen der TU Darmstadt und den vom Fachbereich Informatik am 14.07.2022 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.</p>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning M. Sc. IT Sicherheit  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Noriko Kurachi: The Magic of Computer Graphics. A K Peters/CRC Press Richard Szeliski: Algorithms and Applications, Springer Marcus Magnor, Oliver Grau, Olga Sorkine-Hornung, Christian Theobalt: Digital Representations of the Real World: How to Capture, Model, and Render Visual Reality Wolfgang Förstner, Bernhard P. Wrobel: Photogrammetric Computer Vision - Geometry, Orientation and Reconstruction
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Geometric Algebra Computing					
Modul Nr. 20-00-0490	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Englisch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0490-iv	Geometric Algebra Computing	6	Integrierte Veranstaltung	4
2	Lerninhalt Geometric Computing mit Hilfe einer geometrisch intuitiven Algebra.				
3	Qualifikationsziele / Lernergebnisse Nach Besuch der Veranstaltung sind die Teilnehmenden in der Lage ein neues math. System auf Gebieten wie Visual Computing und Robotik anzuwenden.				
4	Voraussetzung für die Teilnahme				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0490-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%)				
7	Benotung Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0490-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li></ul>				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> 1. Geometric Algebra for Computer Science von Dorst, Fontijne und Mann 2. Dissertation: <a href="http://tuprints.ulb.tu-darmstadt.de/epda/000764/">http://tuprints.ulb.tu-darmstadt.de/epda/000764/</a>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
User-Centered Design in Visual Computing					
<b>Modul Nr.</b> 20-00-0793	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0793-iv	User-Centered Design in Visual Computing	3	Integrierte Veranstaltung	2
<b>2</b>	<b>Lerninhalt</b> Die Entwicklung von benutzerzentrierten Softwarelösungen dient nicht nur zur besseren und effizienteren Nutzung von Software, sie erhöht vielmehr die Akzeptanz und somit auch die Verbreitung und Verwendung. Die Vorlesung “User Centered Design in Visual Computing“ richtet sich in erster Linie an Studierende des Fachbereichs Informatik und vermittelt Modelle, Methoden und Techniken zur benutzerzentrierten Entwicklung von Visualisierungssoftware und visuell-interaktiven Benutzerschnittstellen. Dabei werden insbesondere Methoden vorgestellt, die zu einer gesteigerten Akzeptanz und effizienterer Benutzung der entworfenen Lösungen führen. Des Weiteren werden Methoden der Evaluation vorgestellt, die die Akzeptanz und Nutzbarkeit messen. Die Vorlesung behandelt die eingeführten Themen mit besonderem Bezug zu Visual Computing und graphischen Benutzerschnittstellen. Stoffplan: <ul style="list-style-type: none"><li>• Usability</li><li>• User Experience</li><li>• Task Analysis</li><li>• Benutzerschnittstellen</li><li>• Interaktionsdesign</li><li>• Prototyping</li><li>• Graphikdesign und Informationsvisualisierung</li><li>• Evaluation während und nach der Softwareentwicklung</li><li>• Anwendungen</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende können nach erfolgreichem Besuch der Veranstaltung: <ul style="list-style-type: none"><li>• Geeignete Methoden zur Entwicklung von benutzerzentrierten Softwarelösungen identifizieren und begründen</li><li>• Techniken zu benutzerzentrierten Nutzungsschnittstellen anwenden</li><li>• Evaluationsmethoden zur Untersuchung der eingesetzten Techniken in den verschiedenen Phasen der Entwicklung identifizieren und auswählen</li><li>• Verbesserungen zur Informationsaufnahme und Navigation auf Basis vorhandener Untersuchungen und Evaluationen vorschlagen</li></ul>				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlagen des Visual Computing, wie sie beispielsweise in den Veranstaltungen „Visual Computing“ und „Graphische Datenverarbeitung I,“ vermittelt werden
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0793-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0793-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Hands-On HCI					
<b>Modul Nr.</b> 20-00-1116	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1116-iv	Hands-On HCI	6	Integrierte Veranstaltung	4
<b>2</b>	<b>Lerninhalt</b> Vielleicht haben Sie bereits von Virtual / Augmented Reality, 3D-Druck, am Körper getragenen oder anfassbaren (tangible) Benutzeroberflächen gehört oder diese sogar ausprobiert. Der Bereich Human-Computer-Interaktion (HCI) deckt all diese spannenden Themen ab und bietet die Möglichkeit, neue Prototypen zu bauen und diese in Benutzerstudien auszuprobieren. Wenn Sie Theorie und Praxis im Bereich der HCI verbinden möchten ist dieser Kurs - Hands-On HCI - genau das Richtige für Sie. Das Ziel des Kurses ist es, Sie durch den gesamten Forschungszyklus im Bereich der HCI zu führen. Damit kann dieser Kurs eine Vorbereitung für Ihre zukünftige Bachelor- / Masterarbeit in diesem Bereich sein, sowie einen ersten Baustein auf Ihrem akademischen Weg darstellen.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Abschluss des Moduls können Studierende  - drei Ansätze zur HCI-Forschung voneinander unterscheiden und anwenden. - drei Arten empirischer Untersuchungen unterscheiden. - effektiv eine wissenschaftliche Publikation lesen. - zwischen Arten von HCI-Beiträgen unterscheiden. - Forschungsfragen, Hypothesen und experimentelle Variablen formulieren und definieren. - basierend auf den zuvor erarbeiteten Forschungsfragen ein dazu passendes Studiendesign entwerfen. - eine Studie durchführen und dabei quantitative und qualitative Methoden zur Datensammlung verwenden. - quantitative Daten auf der Basis von statistischen Methoden analysieren, auswerten und interpretieren. - qualitative Daten auf der Basis von Grounded Theory analysieren und interpretieren. - den Peer-Review Prozess verstehen und sowie Reviews für eine wissenschaftliche Publikation schreiben.				

	<ul style="list-style-type: none"> <li>- Evaluationstechniken mit und ohne Nutzern verstehen und anwenden.</li> <li>- die gewonnenen Erkenntnisse als wissenschaftliche Publikation verschriftlichen und vor einem Fachpublikum präsentieren.</li> </ul>
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen wird die vorherige Belegung von „TK2: Human-Computer Interaction“ oder einer vergleichbaren Veranstaltung.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1116-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Klausur (Dauer 60 oder 90 oder 120 Minuten), Mündliche Prüfung (Dauer 15 oder 30 Minuten), Hausübungen (optional: einschließlich Testaten)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1116-iv] (Fachprüfung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Visual Computing**

**Wahlbereich Studienbegleitende Leistungen**

**Praktika, Projektpraktika und ähnliche  
Veranstaltungen**



## Modulbeschreibung

<b>Modulname</b> Serious Games Praktikum					
<b>Modul Nr.</b> 20-00-0236	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0236-pr	Serious Games Praktikum	6	Praktikum	4
2	<b>Lerninhalt</b> In dem Praktikum werden für aktuelle Themen aus dem Bereich Serious Games (beispielsweise für Bildung, Gesundheit und Sport) Konzepte entwickelt und prototypisch realisiert. Die Themen haben jeweils Bezug zur aktuell laufenden Forschung des Fachgebiets, teilweise in Kooperation mit Partnern aus der Games Industrie und/oder Serious Games Anwendern.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Absolvieren der Veranstaltung können die Studierenden eine praktische Aufgabenstellung aus dem „Serious Games“-Umfeld eigenständig bearbeiten sowie die dafür nötige Software konzipieren und prototypisch umsetzen. Außerdem können sie die von ihnen erzielten Ergebnisse einem Publikum unter Anwendung von verschiedenen Präsentationstechniken vorstellen sowie eine dazugehörige Fachdiskussion aktiv bestreiten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Programmierkenntnisse (die Programmiersprache ist jeweils abhängig von Thema und kann teilweise frei gewählt werden).				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0236-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				

	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0236-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Keine
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Visualisierung und Animation von Algorithmen und Datenstrukturen					
<b>Modul Nr.</b> 20-00-0344	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0344-pr	Visualisierung und Animation von Algorithmen und Datenstrukturen	6	Praktikum	4
<b>2</b>	<b>Lerninhalt</b> Im Rahmen des Praktikums beschäftigen wir uns mit der Frage, wie die Dynamik von Algorithmen und Datenstrukturen sinnvoll dargestellt werden kann. Dazu wird die Erstellung solcher Animation praktisch an einem System erprobt.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem Besuch der Veranstaltung sind Studierende in der Lage,  - die zur Verfügung gestellte API zur Animation von Algorithmen anzuwenden. - einen gegebenen Algorithmus auf seine zentralen Elemente zu untersuchen. - jeweils eine Visualisierung für die zentralen Elemente von zwei ausgewählten Algorithmus zu konstruieren. - die erstellten Visualisierungen durch die geeignete Wahl von Parametern zu generalisieren. - kritisch zu beurteilen, ob die gewählte Visualisierung den Lernprozess beim Betrachter unterstützt.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Die Teilnehmer benötigen neben guten Java-Kenntnissen Verständnis für Algorithmen und Datenstrukturen.				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0344-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0344-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum aus Künstlicher Intelligenz					
<b>Modul Nr.</b> 20-00-0412	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0412-pr	Praktikum aus Künstlicher Intelligenz	6	Praktikum	4
2	<b>Lerninhalt</b> Studierende müssen alleine oder in Gruppen ein konkretes praktisches Problem aus dem Bereich der Künstlichen Intelligenz bearbeiten und mit Hilfe von selbst zu entwickelnden oder dem Einsatz von bestehenden Software-Werkzeugen lösen. In Semestern, in denen die Veranstaltung nicht auf diesen Seiten angekündigt wird, besteht oftmals dennoch die Möglichkeit zur Bearbeitung individueller Themen (auf Nachfrage).				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach Bearbeitung dieses Praktikums sind die Studierenden in der Lage <ul style="list-style-type: none"><li>• Einsatzmöglichkeiten von Werkzeugen der künstlichen Intelligenz zu erkennen</li><li>• für gegebene Aufgaben passende Werkzeuge auszuwählen und selbständig einzusetzen</li><li>• den Erfolg des Einsatzes solcher Techniken evaluieren und messen zu können</li></ul>				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Basic knowledge in artificial intelligence				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0412-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				

	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0412-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum Visual Computing					
<b>Modul Nr.</b> 20-00-0418	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0418-pr	Praktikum Visual Computing	6	Praktikum	4
2	<b>Lerninhalt</b> Im Rahmen dieses Praktikums werden ausgewählte Themen aus dem Bereich Visual Computing von den Studierenden bearbeitet und am Ende des Praktikums in einem Vortrag vorgestellt. Die konkreten Themen wechseln von Semester zu Semester und sollten direkt mit einem der Lehrenden angesprochen werden.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Abschluss des Praktikums sind die Studierenden dazu in der Lage, selbständig ein Problem aus dem Bereich des Visual Computings zu analysieren, zu lösen und die Ergebnisse zu bewerten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  praktische Programmierkenntnisse, z. B. in Java, C++  Grundkenntnisse oder Interesse, sich mit Fragestellungen des Visual Computing zu befassen  der Besuch mindestens einer Einführungsvorlesung im Bereich Visual Computing				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0418-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0418-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b> Fortgeschrittenes Praktikum Visual Computing					
<b>Modul Nr.</b> 20-00-0537	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0537-pr	Fortgeschrittenes Praktikum Visual Computing	6	Praktikum	4
2	<b>Lerninhalt</b> Im Rahmen dieses Praktikums werden ausgewählte fortgeschrittene Themen aus dem Bereich Visual Computing von den Studierenden bearbeitet und am Ende des Praktikums in einem Vortrag vorgestellt. Die konkreten Themen wechseln von Semester zu Semester und sollten direkt mit einem der Lehrenden angesprochen werden.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Abschluss des Praktikums sind die Studierenden dazu in der Lage, selbständig ein fortgeschrittenes Problem aus dem Bereich des Visual Computings zu analysieren, zu lösen und die Ergebnisse zu bewerten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  praktische Programmierkenntnisse, z. B. in Java, C++  Grundkenntnisse in Visual Computing  der Besuch mindestens einer Einführungsvorlesung im Bereich Visual Computing sowie „Praktikum Visual Computing“ oder einer vergleichbaren Veranstaltung				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0537-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0537-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

Modulname Advanced User Interfaces					
Modul Nr. 20-00-0570	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0570-pr	Advanced User Interfaces	6	Praktikum	4
2	Lerninhalt <ul style="list-style-type: none"><li>• Analyse von Requirements für eine gegebene Problemstellung</li><li>• Ausarbeitung und Präsentation eines User Interface Konzepts</li><li>• Prototypische Implementierung des Konzepts</li></ul>				
3	Qualifikationsziele / Lernergebnisse Studierende haben einen Einblick in die Prinzipien und Methoden zum Entwurf und zur Entwicklung multimedialer, kollaborativer und adaptiver Benutzungsschnittstellen an Hand einer praktischen Anwendung unter Berücksichtigung verschiedener Kontextbedingungen bekommen.				
4	Voraussetzung für die Teilnahme Empfohlen: <ul style="list-style-type: none"><li>• Interesse an neuen, innovativen Benutzungsschnittstellen</li><li>• Wünschenswert sind Grundkenntnisse der Human Computer Interaction</li><li>• gute Programmierkenntnisse (C#/WPF und/oder Java)</li></ul>				
5	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0570-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%)				
7	Benotung				

	<p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0570-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
<b>9</b>	<p><b>Literatur</b></p> <p>Abhängig von der Aufgabenstellung</p>
<b>10</b>	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Serious Games Projektpraktikum					
<b>Modul Nr.</b> 20-00-0649	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0649-pp	Serious Games Projektpraktikum	9	Projektpraktikum	6
2	<b>Lerninhalt</b> In dem Projektpraktikum werden für aktuelle Themen aus dem Bereich Serious Games (beispielsweise für Bildung, Gesundheit und Sport) Konzepte entwickelt und prototypisch realisiert. Die Themen haben jeweils Bezug zur aktuell laufenden Forschung des Fachgebiets, teilweise in Kooperation mit Partnern aus der Games Industrie und/oder Serious Games Anwendern.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Absolvieren der Veranstaltung können die Studierenden eine praktische Aufgabenstellung aus dem „Serious Games“-Umfeld eigenständig bearbeiten sowie die dafür nötige Software konzipieren und prototypisch umsetzen. Zusätzlich erwerben sie praktisches Wissen im Bereich des Projektmanagements, dass sie nicht nur auf ihr eigenes Thema anwenden, sondern auch auf zukünftige Projekte transferieren können. Außerdem können sie die von ihnen erzielten Ergebnisse einem Publikum unter Anwendung von verschiedenen Präsentationstechniken vorstellen sowie eine dazugehörige Fachdiskussion aktiv bestreiten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Programmierkenntnisse (die Programmiersprache ist jeweils abhängig von Thema und kann teilweise frei gewählt werden).				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00- 0649-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00- 0649-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Keine
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Lernende Roboter: Integriertes Projekt, Teil 1					
<b>Modul Nr.</b> 20-00-0753	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 120 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0753-pj	Lernende Roboter: Integriertes Projekt, Teil 1	6	Praktikum	4
2	<b>Lerninhalt</b> In "Lernende Roboter: Integriertes Projekt, Teil 1" wird zunächst von Studierenden unter Anleitung eine aktuelle Problemstellung des Roboter-Lernens erarbeitet, welche den Forschungsinteressen der Studierenden entspricht, und eine Literaturstudie durchgeführt. Basierend auf diesen Vorarbeiten werden ein Projektplan ausgearbeitet, die notwendigen Algorithmen erprobt und eine prototypische Realisierung in Simulation erstellt.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichen Abschluss der Lehrveranstaltung, können Studierende unabhängig kleine Forschungsprojekte im Bereich Robot Learning aufbauen und in Simulation erproben.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Gleichzeitiger oder vorheriger Besuch der Vorlesung „Lernende Roboter“ oder einer vergleichbaren Veranstaltung.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0753-pj] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
7	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-0753-pj] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b> Projektpraktikum Programmierung Massiv Paralleler Systeme					
<b>Modul Nr.</b> 20-00-0763	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0763-PP	Projektpraktikum Programmierung Massiv Paralleler Systeme	9	Projektpraktikum	6
2	<b>Lerninhalt</b> Im Rahmen dieses Projektpraktikums werden größere ausgewählte Themen aus dem Bereich der Programmierung massiv-paralleler Systeme (wie z.B. GPUs) von den Studierenden in Gruppen bearbeitet und am Ende des Projektpraktikums in einem Vortrag vorgestellt. Die konkreten Themen wechseln von Semester zu Semester.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach dem erfolgreichen Abschluss des Projektpraktikums sind die Studierenden in der Lage große massiv-parallele Projekte zu bearbeiten, welche den Umfang der meisten anderen Projekte während des Studiums weit übersteigen. Hierzu können sie aktuelle Techniken analysieren, modifizieren und anwenden.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: gute C/C++ Programmierkenntnisse, Grundlagen der massiv-parallelen Programmierung (z.B. aus der Veranstaltung „Programmierung Massiv-Paralleler Prozessoren“)				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0763-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0763-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Projektpraktikum Capturing Reality					
<b>Modul Nr.</b> 20-00-0764	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0764-PP	Projektpraktikum Capturing Reality	9	Projektpraktikum	6
2	<b>Lerninhalt</b> Im Rahmen dieses Projektpraktikums werden größere ausgewählte Themen aus dem Bereich Capturing Reality - also an der Schnittstelle von Computer Vision und Computergraphik - von den Studierenden in Gruppen bearbeitet und am Ende des Projektpraktikums in einem Vortrag vorgestellt. Die konkreten Themen wechseln von Semester zu Semester.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Im Rahmen des Projektpraktikums lernen Studierende, eine umfangreiches Problem an der Schnittstelle von Computergraphik und Computer Vision im Team zu lösen. Hierzu können sie aktuelle Techniken analysieren, modifizieren und anwenden.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: empfohlen wird der vorherige Besuch der Veranstaltung „Capturing Reality“ oder einer vergleichbaren Veranstaltung sowie grundlegende Programmierkenntnisse in C/C++				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0764-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>				

	Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0764-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> wird in der Veranstaltung bekanntgegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Projekt-Praktikum Knowledge Engineering und Maschinelles Lernen					
<b>Modul Nr.</b> 20-00-0919	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0919-pp	Projekt-Praktikum Knowledge Engineering und Maschinelles Lernen	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b>  Im Rahmen des Projektpraktikums implementieren Studierende eine vordefinierte, größere Aufgabe aus den Gebieten Knowledge Engineering, Künstliche Intelligenz, maschinelles Lernen und Data Mining. Dabei werden die Themen jeweils in Zusammenarbeit mit dem Betreuer definiert.  Mögliche Themenfelder: - Maschinelles Lernen und Data Mining - Induktives Regel-Lernen - Learning from Preferences - Multilabel Classification - Information Extraction - Web Mining - Semantic Web - Game Playing  Konkrete Aufgabenstellungen werden individuell vereinbart, und das Praktikum kann jederzeit begonnen werden.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach Bearbeitung dieses Projekts sind die Studierenden in der Lage - selbständig größere Programmieraufgaben in den Bereichen Knowledge Engineering, Künstliche Intelligenz, Maschinelles Lernen und Data Mining durchzuführen - mit Hilfe der implementierte Instrumente wissenschaftliche Experimente und Evaluierungen durchzuführen				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Grundlegende Kenntnisse in den Bereichen Knowledge Engineering, Künstliche Intelligenz, Data Mining und maschinelles Lernen sind hilfreich. Es werden außerdem grundlegende Kenntnisse in einer Programmiersprache (z.B. Java o.ä.) vorausgesetzt.				

	Darüber hinaus ist aber besonders die Motivation zur selbstständigen Arbeit und das Interesse an aktuellen Forschungsfragen relevant.
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0919-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0919-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Projektpraktikum Deep Learning in der Computer Vision					
<b>Modul Nr.</b> 20-00-0980	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0980-pp	Projektpraktikum Deep Learning in der Computer Vision	9	Praktikum	6
<b>2</b>	<b>Lerninhalt</b> Im Rahmen des Projektpraktikums werden ausgewählte Themen aus dem Bereich des Deep Learning (tiefe neuronale Netze) für Fragestellungen in der Computer Vision in Gruppen bearbeitet. Dazu gehört die praktische Umsetzung mit modernen Deep Learning Frameworks. Die Ergebnisse werden am Ende in einem Vortrag vorgestellt. Die konkreten Themen orientieren sich am aktuellen Stand der Forschung und wechseln von Semester zu Semester.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Durch erfolgreiche Teilnahme erwerben Studierende vertiefte Kenntnisse in tiefen neuronalen Netzen und deren Anwendungen in der Computer Vision. Sie können aktuelle Techniken in diesem Bereich analysieren, modifizieren und anwenden. Sie trainieren weiterhin Präsentationsfähigkeiten und die Arbeit in einem Team.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  * Gute Programmierkenntnisse in C/C++ oder Python oder Lua  * Vorherige oder parallele Belegung von "Computer Vision I" oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0980-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0980-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

Modulname					
Gründung eines IT-Start-Up					
Modul Nr. 20-00-1016	Leistungspunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-1016-pr	Gründung eines IT-Start-Up	6	Praktikum	4
2	<b>Lerninhalt</b> Kennenlernen von Methoden zur Entwicklung und Umsetzung innovativer Geschäftsmodelle. Erlernen von Werkzeugen für die einzelnen Prozessschritte. Dabei werden Beispiele aus der Praxis vorgestellt und besprochen.  Einüben der vorgestellten Methoden an einem selbstgewählten Beispiel. Präsentation der Ergebnisse nach jedem Teilschritt im Rahmen der Erarbeitung des Geschäftsmodells.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch dieser Veranstaltung haben die Studierenden die Grundlagen für die Erstellung eines Businessplans kennengelernt. Sie sind in der Lage die relevanten Fragestellungen bei der Erstellung von Businessplänen für innovative Geschäftsmodelle zu identifizieren und zu bearbeiten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Der vorherige Besuche von „Software Engineering“ und „Teamprojekt Softwareentwicklung“				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1016-pr] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				

7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1016-pr] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Literaturhinweise werden in der Veranstaltung gegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Anwendung von Reinforcement Learning Methoden					
<b>Modul Nr.</b> 20-00-1048	<b>Leistungspunkte</b> 9 CP	<b>Arbeitsaufwand</b> 270 h	<b>Selbststudium</b> 180 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1048-pp	Anwendung von Reinforcement Learning Methoden	9	Projekt	6
<b>2</b>	<b>Lerninhalt</b> In diesem Projekt lernen Studierende das experimentelle Arbeiten in einem interdisziplinären Team, und bekommen so Einblicke in das wissenschaftliche Arbeiten im Reinforcement Learning. Im Projekt entwickeln in einer Kleingruppen unter Anleitung ein gemeinsames Experiment im Reinforcement Learning basierend auf speziellen Plattformen (Cartpole, Furuta-Pendel, etc), werten dieses aus und schreiben einen Forschungsbericht/Paper.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b>  Praktische Einführung in das wissenschaftliche Arbeiten, Durchführung eines Experimentes von der Forschungsidee bis hin zur Veröffentlichung.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Gleichzeitige Belegung der Vorlesung "Reinforcement Learning: Von Grundlagen zu den Tiefen Ansätzen" oder vorhergehende Belegung von "Lernende Roboter."				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1048-pp] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Bericht (optional: einschließlich der Abgabe von Quellcode), Kolloquium (optional: einschließlich Präsentation)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>  Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-1048-pp] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Projektseminar Autonomes Fahren I					
<b>Modul Nr.</b> 18-su-2070	<b>Leistungspunkte</b> 6 CP	<b>Arbeitsaufwand</b> 180 h	<b>Selbststudium</b> 135 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	18-su-2070-pj	Projektseminar Autonomes Fahren I	0	Projektseminar	3
<b>2</b>	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Praktische Programmiererfahrung mit C++ bei der Entwicklung eingebetteter Systemsoftware aus dem Bereich des autonomen Fahrens anhand eines Modellautos</li><li>• Anwenden von Regelungs- und Steuerungsmethoden aus dem Bereich des autonomen Fahrens</li><li>• Einsatz von Software-Engineering-Techniken (Design, Dokumentation, Test, ...) eines nicht trivialen eingebetteten Software-Systems mit harten Echtzeit-Anforderungen und beschränkten Ressourcen (Speicher, ...)</li><li>• Nutzung eines vorgegebenen Software-Rahmenwerks und Anwendung von weiteren Bibliotheken inklusive eines modular aufgebauten (Echtzeit-)Betriebssystems</li><li>• Einsatz von Source-Code-Management-Systemen, Zeiterfassungswerkzeugen und sonstigen Projektmanagement-Tools</li><li>• Präsentation von Projektergebnissen im Rahmen von Vorträgen</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Studierende sammeln im Rahmen dieses Projektseminars praktische Erfahrung in der Software-Entwicklung für eingebettete Systeme aus dem Bereich des autonomen Fahrens anhand eines Modellautos. Dabei lernen sie in Teamarbeit eine umfangreiche Aufgabe zu bewältigen. Zur Lösung dieser Aufgabe wird geübt, dass in der Gruppe vorhandene theoretische Wissen (aus anderen Lehrveranstaltungen wie Echtzeitsysteme, Software-Engineering - Einführung, C++ Praktikum, Digitale Regelungssysteme) gezielt zur Lösung der praktischen Aufgabe einzusetzen.</p> <p>Studierende, die an diesem Projektseminar erfolgreich teilgenommen haben, sind in der Lage, zu einer vorgegebenen Problemstellung ein größeres Softwareprojekt in einem interdisziplinären Team eigenständig zu organisieren und auszuführen. Die Teilnehmer erwerben folgende Fähigkeiten im Detail:</p> <ul style="list-style-type: none"><li>• Eigenständiges Einarbeiten in ein vorgegebenes Rahmenwerk und vorgefertigten Bibliotheken</li><li>• Umsetzung von theoretischem Wissen in ein Softwaresystem</li><li>• Umfangreicher Einsatz von Werkzeugen zur Versions-, Konfiguration- und Änderungsverwaltung</li><li>• Realistische Zeitplanung und Ressourceneinteilung (Projektmanagement)</li></ul>				

	<ul style="list-style-type: none"> <li>• Entwicklung von Hardware-/Software-Systemen mit C++ unter Berücksichtigung wichtiger Einschränkungen eingebetteter Systeme</li> <li>• Planung und Durchführung umfangreicherer Qualitätssicherungsmaßnahmen</li> <li>• Zusammenarbeit und Kommunikation in und zwischen mehreren Teams</li> </ul>
4	<b>Voraussetzung für die Teilnahme</b> Empfohlene Voraussetzungen sind: <ul style="list-style-type: none"> <li>• ETiT, WI-ETiT (DT), iST, Informatik: Grundlegende Softwaretechnik-Kenntnisse sowie vertiefte Kenntnisse objektorientierter Programmiersprachen (insbesondere: C++)</li> </ul> Zusätzlich erwünscht: <ul style="list-style-type: none"> <li>• Grundlagen der Entwicklung von Echtzeitsystemen oder der Bildverarbeitung</li> <li>• ETiT, WI-ETiT (AUT), MEC: Grundlagen der Regelungstechnik, Reglerentwurf im Zustandsraum, ggf. Grundlagen der digitalen Regelung</li> </ul>
5	<b>Prüfungsform</b> Modulabschlussprüfung: <ul style="list-style-type: none"> <li>• Modulprüfung (Studienleistung, mündliche Prüfung, Dauer 30 Min, Standard)</li> </ul>
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b>
7	<b>Benotung</b> Modulabschlussprüfung: <ul style="list-style-type: none"> <li>• Modulprüfung (Studienleistung, mündliche Prüfung, Gewichtung: 100%)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> MSc ETiT, BSc iST
9	<b>Literatur</b> <a href="https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/ps-af-i/">https://www.es.tu-darmstadt.de/lehre/aktuelle-veranstaltungen/ps-af-i/</a> und Moodle
10	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Visual Computing**

**Wahlbereich Studienbegleitende Leistungen**

**Seminare**

## Modulbeschreibung

<b>Modulname</b> Seminar aus Data Mining und Maschinellem Lernen					
<b>Modul Nr.</b> 20-00-0102	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und English			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0102-se	Seminar aus Data Mining und Maschinellem Lernen	3	Seminar	2
2	<b>Lerninhalt</b> Dieses Seminar dient zur Aufarbeitung neuerer Forschungsarbeiten im Bereich des Data Mining und des Maschinellen Lernens. Für jeden Seminar-Termin wird ein Teilnehmer/eine Teilnehmerin ein Papier vortragen, welches dann von allen Teilnehmenden diskutiert wird. Ausgewählt werden neuere Publikationen aus den relevanten Journalen des Gebiets, insbesondere aus den Journalen "Data Mining and Knowledge Discovery", "Machine Learning", sowie "Journal of Machine Learning Research". Es können aber (nach Rücksprache) auch eigene Themenvorschläge ausgearbeitet werden.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach diesem Seminar sind Studierende in der Lage <ul style="list-style-type: none"><li>• einen unbekannten Text im Bereich des maschinellen Lernens selbständig aufzuarbeiten</li><li>• eine Präsentation für ein Fachpublikum in diesem Gebiet zu entwickeln</li><li>• an einer Fachdiskussion über ein Thema aus dem Gebiet des maschinellen Lernens sinnvoll teilzunehmen</li></ul>				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Basic knowledge in Machine Learning in Data Mining				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0102-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				



	Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0102-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> 3D Animation & Visualisierung					
<b>Modul Nr.</b> 20-00-0216	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0216-se	3D Animation & Visualisierung	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Im Mittelpunkt dieses Seminars stehen aktuelle Arbeiten aus den Themenbereichen physikalisch basierte Simulation, Animation, Echtzeitrendering und Visualisierung. <ul style="list-style-type: none"><li>eigenständiges Einarbeiten in ein Thema anhand von bereitgestellten wissenschaftlichen Arbeiten (i.d.R. englischsprachig)</li><li>Interpretation und Einordnen der Ergebnisse der Literaturarbeit</li><li>Erstellen einer textuellen Zusammenfassung und eines Vortrags über die Thematik</li><li>Präsentation vor einem Publikum mit heterogenem Vorwissen + Fachdiskussion</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Studierenden erhalten in diesem Seminar Fach- und Methodenkompetenz durch die Erarbeitung eines wissenschaftlichen Themas anhand vorgegebener und selbst recherchierter Fachliteratur. Sie können die wesentlichen Aspekte der untersuchten Arbeiten erkennen und diese kompakt aufbereiten, sowohl in textueller als auch in Vortragsform für ein Publikum mit heterogenem Vorwissensstand. Nach dem Vortrag können die Vortragenden aktiv eine Fachdiskussion zu dem von ihnen präsentierten Thema bestreiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: GDV I, (GDV II)				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0216-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.				

	Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0216-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Ausgewählte Artikel von ACM SIGGRAPH, EUROGRPAHICS, IEEE und ähnlichen Konferenzen. Alle Artikel sind in englischer Sprache.  Selected articles from ACM SIGGRAPH, EUROGRPAHICS, IEEE and similar Conferences. All articles are written in English.
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Visual Analytics: Interaktive Visualisierung sehr großer Datenmengen					
<b>Modul Nr.</b> 20-00-0268	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0268-se	Visual Analytics: Interaktive Visualisierung sehr großer Datenmengen	3	Seminar	2
2	<b>Lerninhalt</b> Dieses Seminar richtet sich an Informatiker, die sich für den Bereich der Informationsvisualisierung interessieren, insbesondere den Teilbereich, der sich mit der Visualisierung extrem großer Datenmengen beschäftigt. Die Studenten werden in diesem Seminar eigene Themen im Bereich Visual Analytics erarbeiten, wissenschaftlich aufarbeiten und präsentieren. Zudem wird im Seminar von jedem Teilnehmer ein Aufsatz zum selben Thema ausgearbeitet werden.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch der Veranstaltung haben die Studierenden die Fach- und Methodenkompetenz zur Erarbeitung eines wissenschaftlichen Themas anhand vorgegebener und selbst recherchierter Fachliteratur. Die Studierenden können Themen analysieren, präsentieren und fachlich intensiv diskutieren.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Interesse sich mit einer graphisch-analytischen Fragestellung bzw. Anwendung aus der aktuellen Fachliteratur zu befassen. Vorkenntnisse in Graphischer Datenverarbeitung, Informationssysteme oder Informationsvisualisierung				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0268-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0268-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Serious Games Seminar					
<b>Modul Nr.</b> 20-00-0328	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 120 h	<b>Selbststudium</b> 90 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0328-se	Serious Games Seminar	4	Seminar	2
2	<b>Lerninhalt</b> In dem Seminar wird der aktuelle Stand der Forschung bezüglich des Einsatzes von Serious Games (beispielsweise für Bildung, Gesundheit und Sport) analysiert und diskutiert. Die Themen haben jeweils Bezug zur aktuell laufenden Forschung des Fachgebiets, teilweise in Kooperation mit Partnern aus der Games Industrie und/oder Serious Games Anwendern.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an dem Seminar können sich die Studierenden eigenständig in ein Thema aus dem Bereich „Serious Games“ einarbeiten. Sie sind mit Techniken der Literaturrecherche im Bereich von wissenschaftlichen Veröffentlichungen und von Industriequellen vertraut. Die dort genannten Techniken bzw. Ergebnisse können von ihnen zusammengefasst, bewertet und untereinander verglichen werden. Außerdem können sie die von ihnen erzielten Ergebnisse einem Publikum unter Anwendung von verschiedenen Präsentationstechniken vorstellen sowie eine dazugehörige Fachdiskussion aktiv bestreiten.				
4	<b>Voraussetzung für die Teilnahme</b>				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0328-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0328-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Keine
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Aktuelle Trends im Medical Computing					
<b>Modul Nr.</b> 20-00-0468	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0468-se	Aktuelle Trends im Medical Computing	3	Seminar	2
2	<b>Lerninhalt</b> - Selbstständiges Studium aktueller Conference und Journal Papers aus dem Bereich Medical Imaging zu einem ausgewählten Thema im Bereich grundlegender Methoden. - Kritische Auseinandersetzung mit dem behandelten Thema - Eigene weiterführende Literaturrecherchen - Erstellen eines Vortrags (schriftliche Ausarbeitung und Folienpräsentation) über die behandelte Thematik - Präsentation des Vortrags vor Publikum mit heterogenem Vorwissen - Fachliche Diskussion über die behandelte Thematik nach dem Vortrag - Medizinische Anwendungsfelder sind u.a. Onkologie, Orthopädie, navigierte Chirurgie Behandelte Methoden umfassen u.a.: Segmentierung, Registrierung, Visualisierung, Simulation, Navigation und Tracking.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden sich eigenständig in ein Thema anhand von wissenschaftlichen Veröffentlichungen einarbeiten.  Sie lernen die wesentlichen Aspekte der untersuchten Arbeiten zu erkennen und auf verständliche Weise einem heterogenen Publikum vorzutragen. Dabei wenden sie verschiedene Präsentationstechniken an. Nach dem Vortrag können die Studierenden aktiv eine Fachdiskussion zu dem präsentierten Thema leiten und bestreiten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Bachelor ab 4. Semester, Master ab 1. Semester.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0468-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul>				



	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0468-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Wird zu Beginn der Veranstaltung bekanntgegeben.</p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Skalenraum- und PDE-Methoden in der Bildanalyse und -verarbeitung					
<b>Modul Nr.</b> 20-00-0469	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0469-se	Skalenraum- und PDE-Methoden in der Bildanalyse und -verarbeitung	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Bildanalyse und -verarbeitung beschäftigen sich mit der Untersuchung von Bildern und der Anwendung bestimmter Aufgaben auf Bilder, wie Verbesserung, Rauschunterdrückung, Schärfung und Segmentierung. In diesem Kurs werden häufig verwendete mathematische Methoden vorgestellt und diskutiert. Der Schwerpunkt liegt dabei auf der axiomatischen Modellwahl, deren mathematischen Eigenschaften und dem praktischen Nutzen. Stichwörter: <ul style="list-style-type: none"><li>• Filterung (Kantenerkennung, Verbesserung, Wiener, Fourier, ...)</li><li>• Bilder &amp; Beobachtungen: Skalenraum, Regularisierung, Distributionen</li><li>• Objekte: Differenzstruktur, Invarianten, Feature-Erkennung</li><li>• Tiefenstruktur: Katastrophen und Multi-Skalen-Hierarchie</li><li>• Variationsmethoden und Partielle Differentialmethoden: Perona-Malik, anisotrope Diffusion, Total Variation, Mumford-Shah, Chan-Vese, geometrische partielle Differentialgleichungen, Level-Sets.</li><li>• Kurvenevolution: Normalenbewegung, mittlere Krümmungsbewegung, euklidische Verkürzungsbewegung.</li></ul>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreichem Besuch der Veranstaltung beschreiben Studierende die Grundkonzepte sowie grundlegenden mathematische Modelle und Methoden der Bildanalyse und -verarbeitung. Sie erklären wichtige Verfahren zu Skalenraum- sowie zu PDE-Ansätzen und können damit repräsentative Fachbeiträge beschreiben, beurteilen, und transferieren.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Da Bildanalyse und -verarbeitung eine Mischung aus verschiedenen Disziplinen, wie Physik, Mathematik, Vision, Informatik und Engineering, ist, ist dieser Kurs gezielt auf ein breites Publikum zugeschnitten. Daher werden nur Grundkenntnisse in Analysis				

	angenommen. Weitere notwendige mathematische Werkzeuge werden in den Sitzungen skizziert.
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0469-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> <p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0469-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Main: <ul style="list-style-type: none"> <li>B. M. ter Haar Romeny, Front-End Vision and Multi-scale Image Analysis, Dordrecht, Kluwer Academic Publishers, 2003.</li> </ul> Recommended: <ul style="list-style-type: none"> <li>T. Lindeberg: Scale-Space Theory in Computer Vision, Dordrecht, Kluwer Academic Publishers, 1994.</li> <li>J. Weickert: Anisotropic Diffusion in Image Processing, Teubner-Verlag, Stuttgart, Germany, 1998.</li> <li>G. Aubert &amp; P. Kornprobst: Mathematical problems in image processing: Partial Differential Equations and the Calculus of Variations (second edition), Springer, Applied Mathematical Sciences, Vol 147, 2006.</li> </ul>
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname Semantik Visualisierung					
Modul Nr. 20-00-0542	Leistungspunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch und English			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0542-se	Semantik Visualisierung	3	Seminar	2
2	<b>Lerninhalt</b> Eigenständige wissenschaftliche Ausarbeitung eines in der Forschung aktuellen Themas aus dem Bereich Semantik Visualisierung • Eigene Literaturrecherchen, angeleitet von Betreuer • Interpretation und Einordnen der Ergebnisse der Literaturarbeit, zusammen mit Betreuer • Erstellen einer schriftlichen Ausarbeitung zu dem gewählten Thema (Deutsch oder Englisch), angeleitet vom Betreuer • Erstellen eines Vortrages zu der ausgearbeiteten Thematik, angeleitet von Betreuer • Halten des Vortrages vor einem Fachpublikum Feedback an die Vortragenden zu den Vorträgen (u.a. betreffend Rhetorik, Präsentationstechniken) und zur Fachdiskussion				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden sich eigenständig in ein Thema anhand von wissenschaftlichen Veröffentlichungen einarbeiten und dieses wissenschaftlich beschreiben. Sie lernen diverse Techniken und Ressourcen der Literaturrecherche kennen und können diese auch für weitere Arbeiten einsetzen. Des Weiteren werden die Studierenden mit praktischen und aktuellen Themen aus der angewandten Forschung konfrontiert und lernen dabei interessante Themengebiete kennen.  Die schriftliche Ausarbeitung ermöglicht die wissenschaftliche Wiedergabe in Form von Schrift, während die Präsentation vor einem Fachpublikum die mündliche Wiedergabe fördert. Somit lernen die Studierenden ein Thema zu recherchieren, schriftlich auszuarbeiten und zu präsentieren.  Nach dem Vortrag können die Vortragenden aktiv eine Fachdiskussion zu dem von ihnen präsentierten Thema bestreiten.				

4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Besuch der Vorlesung Visual Computing oder einer vergleichbaren Veranstaltung
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0542-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0542-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Wird jeweils passend zu den aktuellen Themen bekanntgegeben
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Fortgeschrittene Themen in der Computergraphik					
<b>Modul Nr.</b> 20-00-0604	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0604-se	Fortgeschrittene Themen in der Computergraphik	3	Seminar	2
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Grundlagen der wissenschaftlichen Vortragstechnik und Begutachtung</li><li>• Eigenständiges Einarbeiten in aktuelle Publikationen in Computergraphik (englischsprachig)</li><li>• Eigene darüber hinausgehende Recherche zur Hintergrund-Literatur, angeleitet von Betreuer</li><li>• Erstellen eines zweiteiligen Vortrags (Problemstellung und Lösungsansatz) über eine Publikationen einschließlich Folienpräsentation, angeleitet durch Betreuer</li><li>• Erstellen eines (simulierten) wissenschaftlichen Gutachtens über eine zweite Publikation, angeleitet durch Betreuer</li><li>• Halten des Vortrags vor einem Publikum mit heterogenem Vorwissen</li><li>• Führung der Fachdiskussion nach beiden Vortragsteilen</li><li>• Aktive Teilnahme an den Fachdiskussionen, sowie Feedback an die Vortragenden</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden sich eigenständig in aktuelle Themen der Computergraphik anhand von wissenschaftlichen Veröffentlichungen einarbeiten. Sie können die wesentlichen Beiträge der untersuchten Publikationen erkennen und diese kompakt einem Publikum mit heterogenem Vorwissensstand präsentieren, unter Berücksichtigung von Prinzipien des guten wissenschaftlichen Vortrags. Nach dem Vortrag können die Vortragenden aktiv eine Fachdiskussion zu dem von ihnen präsentierten Thema bestreiten. Weiterhin sind sie in der Lage ein wissenschaftliches Gutachten über eine aktuelle Publikation anzufertigen, welches den üblichen Standards des wissenschaftlichen Begutachtungsprozesses genügt.</p>				
4	<b>Voraussetzung für die Teilnahme</b>				

	Empfohlen: Teilnehmer sollten Grundkenntnisse in Computergraphik besitzen (z.B. durch Besuch von Graphische Datenverarbeitung I).
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0604-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen. Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0604-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> aktuelle Publikationen, überwiegend des vergangenen Jahres
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Fortgeschrittene Themen in Computer Vision und Maschinellem Lernen					
<b>Modul Nr.</b> 20-00-0645	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0645-se	Fortgeschrittene Themen in Computer Vision und Maschinellem Lernen	3	Seminar	2
2	<b>Lerninhalt</b> <ul style="list-style-type: none"><li>• Grundlagen der wissenschaftlichen Vortragstechnik und Begutachtung</li><li>• Eigenständiges Einarbeiten in aktuelle Publikationen in Computer Vision oder Maschinellem Lernen (englischsprachig)</li><li>• Eigene darüber hinausgehende Recherche zur Hintergrund-Literatur, angeleitet von Betreuer</li><li>• Erstellen eines zweiteiligen Vortrags (Problemstellung und Lösungsansatz) über eine Publikationen einschließlich Folienpräsentation, angeleitet durch Betreuer</li><li>• Erstellen eines (simulierten) wissenschaftlichen Gutachtens über eine zweite Publikation, angeleitet durch Betreuer</li><li>• Halten des Vortrags vor einem Publikum mit heterogenem Vorwissen</li><li>• Führung der Fachdiskussion nach beiden Vortragsteilen</li><li>• Aktive Teilnahme an den Fachdiskussionen, sowie Feedback an die Vortragenden</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden sich eigenständig in aktuelle Themen der Computer Vision und/oder des Maschinellen Lernens anhand von wissenschaftlichen Veröffentlichungen einarbeiten. Sie können die wesentlichen Beiträge der untersuchten Publikationen erkennen und diese kompakt einem Publikum mit heterogenem Vorwissensstand präsentieren, unter Berücksichtigung von Prinzipien des guten wissenschaftlichen Vortrags. Nach dem Vortrag können die Vortragenden aktiv eine Fachdiskussion zu dem von ihnen präsentierten Thema bestreiten. Weiterhin sind sie in der Lage ein wissenschaftliches Gutachten über eine aktuelle Publikation anzufertigen, welches den üblichen Standards des wissenschaftlichen Begutachtungsprozesses genügt.</p>				



4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Teilnehmer sollten Grundkenntnisse in Computer Vision, sowie idealerweise maschinellem Lernen besitzen (z.B. durch Besuch von Computer Vision I, Statistisches Maschinelles Lernen).
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0645-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0645-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b> Aktuelle Publikationen, überwiegend des vergangenen Jahres
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Computergestützte Planung und Navigation in der Medizin					
<b>Modul Nr.</b> 20-00-0677	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0677-se	Computergestützte Planung und Navigation in der Medizin	3	Seminar	2
2	<b>Lerninhalt</b> - Selbstständiges Studium aktueller Konferenz- und Journal-Papers aus dem Bereich Medical Imaging zu einem ausgewählten Thema im Bereich der Planung und chirurgischen Navigation. - Kritische Auseinandersetzung mit dem behandelten Thema - Eigene weiterführende Literaturrecherchen - Erstellen eines Vortrags (schriftliche Ausarbeitung und Folienpräsentation) über die behandelte Thematik - Präsentation des Vortrags vor Publikum mit heterogenem Vorwissen - Fachliche Diskussion über die behandelte Thematik nach dem Vortrag Behandelte Methoden, die in Zusammenhang mit der Operationsplanung und navigierten Chirurgie stehen sind u.a.: Segmentierung, Registrierung, Visualisierung, Simulation, Navigation und Tracking.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Nach erfolgreicher Teilnahme an der Veranstaltung können die Studierenden sich eigenständig in ein Thema anhand von wissenschaftlichen Veröffentlichungen einarbeiten.  Sie lernen die wesentlichen Aspekte der untersuchten Arbeiten zu erkennen und auf verständliche Weise einem heterogenen Publikum vorzutragen. Dabei wenden sie verschiedene Präsentationstechniken an. Nach dem Vortrag können die Studierenden aktiv eine Fachdiskussion zu dem präsentierten Thema leiten und bestreiten.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Bachelor ab 4. Semester, Master ab 1. Semester.				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0677-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)				

	<p>Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.</p> <p>Kolloquium (optional: einschließlich Präsentation), Hausarbeit</p>
6	<p><b>Voraussetzung für die Vergabe von Leistungspunkten</b></p> <p>Bestehen der Prüfung (100%)</p>
7	<p><b>Benotung</b></p> <p>Bausteinbegleitende Prüfung:</p> <ul style="list-style-type: none"> <li>• [20-00-0677-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<p><b>Verwendbarkeit des Moduls</b></p> <p>B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science</p> <p>Kann in anderen Studiengängen verwendet werden.</p>
9	<p><b>Literatur</b></p> <p>Wird zu Beginn der Veranstaltung bekanntgegeben.</p>
10	<p><b>Kommentar</b></p>

## Modulbeschreibung

<b>Modulname</b> Angewandte Themen der Computergraphik					
<b>Modul Nr.</b> 20-00-0724	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0724-se	Angewandte Themen der Computergraphik	3	Seminar	2
2	<b>Lerninhalt</b> Ausgewählte aktuelle Arbeiten aus Forschung und Literatur werden zur Bearbeitung ausgegeben. Die Arbeiten stammen aus folgenden Feldern der Computergraphik: <ul style="list-style-type: none"><li>• Visualisierung / Rendering</li><li>• Simulation</li><li>• Geometrieverarbeitung und Modellierung</li><li>• Semantik und 3D</li></ul>				
3	<b>Qualifikationsziele / Lernergebnisse</b> Studierende kennen nach erfolgreichem Besuch der Veranstaltung einen Ausschnitt an aktuellen Themen der Computergraphik. Sie können sich selbständig den Inhalt einer Veröffentlichung erarbeiten, die Problemstellung und den Lösungsansatz erkennen und präsentieren. Weiter sind sie in der Lage, Verbesserungspotenzial in den Arbeiten zu analysieren und darzustellen.				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Vorkenntnisse in GDV oder Geom. Methoden CAD/CAE sind vorteilhaft				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>• [20-00-0724-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				

<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-0724-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b> Wird in der Veranstaltung bekanntgegeben.
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Algorithmen und Plattformen des Reinforcement Learning					
<b>Modul Nr.</b> 20-00-1050	<b>Leistungspunkte</b> 3 CP	<b>Arbeitsaufwand</b> 90 h	<b>Selbststudium</b> 60 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1050-se	Algorithmen und Plattformen des Reinforcement Learning	3	Seminar	2
<b>2</b>	<b>Lerninhalt</b> Im Rahmen dieses Seminars werden wir Reinforcement Learning Algorithmen und deren Anwendung in Intelligenten Technischen Systemen diskutieren. Hierbei sollen Studenten die Fähigkeit erwerben, sich einen unbekannten Text selbstständig zu erarbeiten, für eine Präsentation aufzubereiten und einem Fachpublikum zu präsentieren.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Am Ende dieses Kurses verstehen Studierende die aktuellen Forschungsthemen im Reinforcement Learning und sind in der Lage die Literaturvorstudie für eine Forschungsarbeit in diesem Bereich durchzuführen.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Gleichzeitige Belegung der Vorlesung "Reinforcement Learning: Von Grundlagen zu den Tiefen Ansätzen" oder vorhergehende Belegung von "Lernende Roboter."				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1050-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>[20-00-1050-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Autonome Systeme und Robotik  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>					
Erweitertes Seminar - Systems and Machine Learning					
<b>Modul Nr.</b> 20-00-1057	<b>Leistungspunkte</b> 4 CP	<b>Arbeitsaufwand</b> 120 h	<b>Selbststudium</b> 75 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1057-se	Erweitertes Seminar - Systems and Machine Learning	4	Seminar	3
<b>2</b>	<b>Lerninhalt</b> <p>Dieses Seminar dient der Diskussion neuer Forschungspapiere im Zusammenhang von Hardware-/Softwaresystemen und maschinellem Lernen (ML). Das Seminar zielt auf die Verbindungen zwischen diesen Themenbereichen ab und diskutiert Fragestellungen, die auf praktisch anwendbares maschinelles Lernen zugeschnitten sind wie z.B. Hardware-Beschleuniger für ML, verteilte skalierbare ML-Systeme, neuer Programmierparadigmen für ML, Automatisiertes ML, sowie Anwendungen von ML für Systeme.</p> <p>Jeder Teilnehmer/jede Teilnehmerin präsentiert ein Forschungspapier, das anschließend von allen Teilnehmenden diskutiert wird. Darüber hinaus werden zusammenfassende Arbeiten in Gruppen verfasst und einem Peer-Review Prozess unterzogen. Die vorzustellenden Arbeiten stellen in der Regel aktuelle Publikationen in relevanten Konferenzen und Zeitschriften dar.</p> <p>Das Seminar wird als Blockveranstaltung angeboten.</p>				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> <p>Nach diesem Seminar sind Studierende in der Lage:</p> <ul style="list-style-type: none"><li>- einen unbekannten Text aus den Bereichen des Seminars selbständig aufzuarbeiten</li><li>- eine Präsentation und eine schriftliche Zusammenfassung für ein Fachpublikum in diesem Gebiet zu entwickeln</li><li>- an einer Fachdiskussion über ein Thema aus den Bereichen des Seminars sinnvoll teilzunehmen</li><li>- die Meinung über eine wissenschaftliche Arbeit in der Form eines schriftlichen Peer-Reviews zu artikulieren</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> <p>Empfohlen: Grundkenntnisse in maschinellem Lernen, skalierbarem Datenmanagement und Hardware-/Softwaresystemen.</p>				



<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1057-se] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li> </ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Hausarbeit
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1057-se] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

**Modulhandbuch**  
**M. Sc. Computer Science**

**Vertiefung Visual Computing**

**Wahlbereich Studienbegleitende Leistungen**

**Praktikum in der Lehre**

## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Visual Computing					
<b>Modul Nr.</b> 20-00-0519	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0519-pl	Praktikum in der Lehre - Visual Computing	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Mitarbeit in der Ausrichtung der Lehrveranstaltung Visual Computing (Übungskonzeption, Korrektur, Begleitung des Lernenden)				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Es sollen Fähigkeiten erlernt werden, geeignete Lernmaterialien für Schulungen in Informatikthemen selbst zu erstellen, ihren Einsatz kritisch zu begleiten und dabei auch die Lernenden zu betreuen und anzuleiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen:  Der vorherige Besuch der Veranstaltung „Visual Computing“ oder einer vergleichbaren Veranstaltung				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-0519-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:				

	<ul style="list-style-type: none"> <li>• [20-00-0519-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum in der Lehre - Informationsvisualisierung und Visual Analytics					
<b>Modul Nr.</b> 20-00-0767	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-0767-pl	Praktikum in der Lehre - Informationsvisualisierung und Visual Analytics	5	Praktikum in der Lehre	3
2	<b>Lerninhalt</b> Dieser Kurs ermöglicht es Studierenden, die Haus- und Präsenzübungen für die Vorlesung “Informationsvisualisierung und Visual Analytics” unter Anleitung durch die Lehrenden zu konzipieren, durchzuführen und die Lernergebnisse der Vorlesungsteilnehmer zu evaluieren.				
3	<b>Qualifikationsziele / Lernergebnisse</b> Die Studierenden können nach erfolgreicher Durchführung der Veranstaltung: •Lehrinhalte aus der Vorlesung für Haus- und Präsenzübungen aufbereiten •Übungen mit Studentengruppen konzipieren und durchführen •Ein Konzept für aufeinander aufbauende praktische Übungen entwickeln •Methoden der Lernkontrolle für die Lerninhalte der Vorlesung anwenden				
4	<b>Voraussetzung für die Teilnahme</b> Empfohlen: Informationsvisualisierung und Visual Analytics				
5	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-0767-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
6	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				

7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>[20-00-0767-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

Modulname					
Praktikum in der Lehre - Graphische Datenverarbeitung II					
Modul Nr. 20-00-0954	Leistungspunkte 5 CP	Arbeitsaufwand 150 h	Selbststudium 105 h	Moduldauer 1 Semester	Angebotsturnus Jedes 2. Semester
Sprache Deutsch			Modulverantwortliche Person Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
1	Kurse des Moduls				
	Kurs Nr.	Kursname	Arbeitsaufwand (CP)	Lehrform	SWS
	20-00-0954-pl	Praktikum in der Lehre - Graphische Datenverarbeitung II	5	Praktikum in der Lehre	3
2	Lerninhalt Das Erstellen von Lehrmaterial, die Beurteilung und Betreuung von Übungen.				
3	Qualifikationsziele / Lernergebnisse Aufbereitung und Vermittlung des Vorlesungsinhaltes.				
4	Voraussetzung für die Teilnahme Empfohlen: Der Vorlesungsinhalt von „Graphische Datenverarbeitung II“				
5	Prüfungsform Bausteinbegleitende Prüfung:  • [20-00-0954-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
6	Voraussetzung für die Vergabe von Leistungspunkten Bestehen der Prüfung (100%)				
7	Benotung Bausteinbegleitende Prüfung:  • [20-00-0954-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)				

<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>



## Modulbeschreibung

<b>Modulname</b>					
Praktikum in der Lehre - Algorithmenvisualisierung					
<b>Modul Nr.</b> 20-00-1036	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1036-pl	Praktikum in der Lehre - Algorithmenvisualisierung	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Im Rahmen des Praktikums in der Lehre Algorithmenvisualisierung werden ein Animationssystem weiterentwickelt, die Aufgabenstellungen des assoziierten Praktikums überarbeitet und die studentischen Abgaben im Rahmen des Praktikums betreut.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Die Teilnehmer am Praktikum in der Lehre Algorithmenvisualisierung vertiefen ihre Kenntnisse im Bereich der Visualisierung sowie der Algorithmen und Datenstrukturen. Gleichzeitig erhalten sie Einblicke in die Lehrtätigkeit durch Betreuung von Studierenden, Bewertung von Aufgaben und Überarbeitung von Aufgaben.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b>				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1036-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				

7	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1036-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
8	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
9	<b>Literatur</b>
10	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b> Praktikum in der Lehre - Graphische Datenverarbeitung I					
<b>Modul Nr.</b> 20-00-1101	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes 2. Semester
<b>Sprache</b> Deutsch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
	20-00-1101-pl	Praktikum in der Lehre - Graphische Datenverarbeitung I	5	Praktikum in der Lehre	3
<b>2</b>	<b>Lerninhalt</b> Das Erstellen von Lehrmaterial, die Beurteilung und Betreuung von Übungen.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Es sollen Fähigkeiten erlernt werden, geeignete Lernmaterialien für Schulungen in Informatikthemen selbst zu erstellen, ihren Einsatz kritisch zu begleiten und dabei auch die Lernenden zu betreuen und anzuleiten.				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen wird:  - Besuch der Veranstaltung "Graphische Datenverarbeitung I" mit sehr guter Abschlussnote - Programmierkenntnisse in C++ und OpenGL				
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung:  • [20-00-1101-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)  Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				

<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> <li>• [20-00-1101-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

## Modulbeschreibung

<b>Modulname</b>						
Praktikum in der Lehre - Visuelle Inferenz						
<b>Modul Nr.</b> 20-00-1131	<b>Leistungspunkte</b> 5 CP	<b>Arbeitsaufwand</b> 150 h	<b>Selbststudium</b> 105 h	<b>Moduldauer</b> 1 Semester	<b>Angebotsturnus</b> Jedes Semester	
<b>Sprache</b> Deutsch und Englisch			<b>Modulverantwortliche Person</b> Koordinatoren/Koordinatorinnen Vertiefung Visual Computing			
<b>1</b>	<b>Kurse des Moduls</b>					
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>	
	20-00-1131-pl	Praktikum in der Lehre - Visuelle Inferenz	5	Praktikum in der Lehre	3	
<b>2</b>	<b>Lerninhalt</b> Erstellung von Übungs- und Vorlesungsmaterial zu Lehrveranstaltungen des FG Visuelle Inferenz					
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse</b> Studierende sind in der Lage: -Lehrinhalte in Übungen zu präsentieren und zu erklären -Praktikumsgruppen zu betreuen -Methoden zur Kontrolle des Lernerfolgs systematisch anzuwenden					
<b>4</b>	<b>Voraussetzung für die Teilnahme</b> Empfohlen wird die erfolgreiche Teilnahme an den Vorlesungen „Computer Vision“ und/oder „Computer Vision II“ oder vergleichbaren Veranstaltungen, je nach Semester.					
<b>5</b>	<b>Prüfungsform</b> Bausteinbegleitende Prüfung: <ul style="list-style-type: none"><li>[20-00-1131-pl] (Studienleistung, mündliche / schriftliche Prüfung, Standard)</li></ul> Die Form der Prüfung wird zu Beginn der Lehrveranstaltung bekannt gegeben. Möglich ist eine oder eine Kombination von maximal zwei der nachfolgend aufgeführten Formen.  Kolloquium (optional: einschließlich Präsentation), Portfolio, Bericht (Optional: einschließlich der Abgabe von Lehrmaterial)					
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%).					
<b>7</b>	<b>Benotung</b> Bausteinbegleitende Prüfung:					

	<ul style="list-style-type: none"> <li>• [20-00-1131-pl] (Studienleistung, mündliche / schriftliche Prüfung, Gewichtung: 100%, Standard)</li> </ul>
<b>8</b>	<b>Verwendbarkeit des Moduls</b> B. Sc. Informatik M. Sc. Informatik M. Sc. Computer Science M. Sc. Artificial Intelligence and Machine Learning  Kann in anderen Studiengängen verwendet werden.
<b>9</b>	<b>Literatur</b>
<b>10</b>	<b>Kommentar</b>

# **Modulhandbuch M. Sc. Computer Science**

## **Masterarbeit**

## Modulbeschreibung

<b>Modulname</b> Masterarbeit Computer Science					
<b>Modul Nr.</b> 20-AM-XXXX	<b>Leistungspunkte</b> 30 CP	<b>Arbeitsaufwand</b> 900 h	<b>Selbststudium</b> 900 h	<b>Moduldauer</b>	<b>Angebotsturnus</b> Jedes Semester
<b>Sprache</b> Deutsch/Englisch			<b>Modulverantwortliche Person</b> Studiendekan/Studiendekanin		
<b>1</b>	<b>Kurse des Moduls</b>				
	<b>Kurs Nr.</b>	<b>Kursname</b>	<b>Arbeitsaufwand (CP)</b>	<b>Lehrform</b>	<b>SWS</b>
<b>2</b>	<b>Lerninhalt</b> Selbständige Bearbeitung einer wissenschaftlichen Fragestellung der Informatik nach wissenschaftlichen Grundsätzen in begrenzter Zeit. Die Problemstellung, Vorgehensweise sowie die Ergebnisse werden schriftlich dokumentiert und mündlich in einem Kolloquium präsentiert.				
<b>3</b>	<b>Qualifikationsziele / Lernergebnisse / Kompetenzen</b> Die Studierenden sind nach der Masterarbeit in der Lage, <ul style="list-style-type: none"><li>• eine komplexere wissenschaftliche Fragestellung mit Forschungsbezug nach wissenschaftlichen Grundsätzen selbstständig zu bearbeiten,</li><li>• die im Studium erworbenen Kenntnisse, Methoden und Kompetenzen zu verknüpfen und anzuwenden,</li><li>• die relevante Literatur zu recherchieren, einzugrenzen und auszuwerten,</li><li>• das Thema sinnvoll zu systematisieren und einen Argumentationsstrang aufzubauen,</li><li>• die Validität von Pro- und Kontraargumenten nachvollziehbar abzuwägen,</li><li>• die Ergebnisse in die aktuelle Forschung einzuordnen und zu bewerten,</li><li>• die Ergebnisse schriftlich nach wissenschaftlichen Grundsätzen niederzulegen,</li><li>• die Ergebnisse zu präsentieren und argumentativ zu vertreten.</li></ul>				
<b>4</b>	<b>Voraussetzung für die Teilnahme</b>				
<b>5</b>	<b>Prüfungsform</b> Thesis				
<b>6</b>	<b>Voraussetzung für die Vergabe von Leistungspunkten</b> Bestehen der Prüfung (100%)				
<b>7</b>	<b>Benotung</b> Standard (Ziffernote)				
<b>8</b>	<b>Verwendbarkeit des Moduls</b> M.Sc. Computer Science				



9	<b>Literatur</b> - Sandberg, Berit: Wissenschaftlich Arbeiten von Abbildung bis Zitat: Lehr- und Übungsbuch für Bachelor, Master und Promotion. De Gruyter Oldenbourg; Auflage: 3, 2017 Ergänzt durch Literatur entsprechend dem Themengebiet der Abschlussarbeit.
10	<b>Kommentar</b> Die Abschlussarbeit muss innerhalb von 26 Wochen angefertigt und eingereicht werden. Sie hat einen Arbeitsaufwand von 900 Stunden. Ein Studium in Regelstudienzeit setzt voraus, dass bei Beginn der Masterarbeit im 4. Semester bei voller Ausschöpfung der Bearbeitungszeit von 26 Wochen nicht später als Anfang Februar bei Studienbeginn zum Jedes 2. Semester bzw. Anfang August bei Studienbeginn zum Jedes 2. Semester begonnen werden muss.