

## AMTLICHE MITTEILUNGEN

VERKÜNDUNGSBLATT DER UNIVERSITÄT PADERBORN AM.UNI.PB

AUSGABE 58.25 VOM 4. JULI 2025

---

# SATZUNG ZUR ÄNDERUNG DER BESONDEREN BESTIMMUNGEN DER PRÜFUNGSORDNUNG FÜR DEN BACHELORSTUDIENGANG LEHRAMT AN GYMNASIEN UND GESAMTSCHULEN MIT DEM UNTERRICHTSFACH INFORMATIK AN DER UNIVERSITÄT PADERBORN

VOM 4. JULI 2025

**Satzung zur Änderung der Besonderen Bestimmungen der Prüfungsordnung für den  
Bachelorstudiengang Lehramt an Gymnasien und Gesamtschulen mit dem  
Unterrichtsfach Informatik an der Universität Paderborn**

**vom 4. Juli 2025**

Aufgrund des § 2 Abs. 4 und des § 64 Abs. 1 des Gesetzes über die Hochschulen des Landes Nordrhein-Westfalen (Hochschulgesetz – HG) vom 16. September 2014 (GV.NRW. S. 547), zuletzt geändert durch Artikel 2 des Gesetzes vom 19. Dezember 2024 (GV. NRW. S. 1222), hat die Universität Paderborn die folgende Satzung erlassen:

**Artikel I**

Die Besonderen Bestimmungen der Prüfungsordnung für den Bachelorstudiengang Lehramt an Gymnasien und Gesamtschulen mit dem Unterrichtsfach Informatik an der Universität Paderborn vom 31. Mai 2022 (AM.Uni.Pb 75.22) werden wie folgt geändert:

1. § 36 wird wie folgt gefasst:  
Das Studienvolumen des Unterrichtsfaches Informatik umfasst 72 Leistungspunkte (LP), davon sind 8 LP fachdidaktische Studien nachzuweisen. 3 LP entfallen auf inklusionsorientierte Fragestellungen.
2. § 38 Absatz 1 wird wie folgt gefasst:  
(1) Das Studienangebot im Umfang von 72 LP, davon 8 LP fachdidaktische Studien, umfasst elf Module.
3. § 38 Absatz 3 wird wie folgt gefasst:  
(3) Die Studierenden erwerben die in § 37 genannten Kompetenzen im Rahmen folgender Module:

<b>B1 Programmierung I</b>		<b>8 LP</b>	
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
1. Sem.	Programmierung I	P	240
<b>B2 Modellierung</b>		<b>8 LP</b>	
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
1. Sem.	Modellierung	P	240
<b>B3 Datenbanksysteme</b>		<b>6 LP</b>	
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
2. Sem.	Datenbanksysteme	P	180

<b>B4 Einführung in Computer Systeme</b>			<b>6 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
2. Sem.	Einführung in Computer Systeme	P	180
<b>B5 Datenstrukturen, Algorithmen und formale Sprachen</b>			<b>8 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
3. Sem.	Datenstrukturen, Algorithmen und formale Sprachen	P	240
<b>B6 Didaktische Grundlagen des Informatikunterrichts – GyGe</b>			<b>8 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
3.-4. Sem.	a) Fachdidaktische Grundlagen b) Stufenbezogene Unterrichtsmodelle – GyGe	P P	240
<b>B7 Programmierung II</b>			<b>6 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
4. Sem.	Programmierung II	P	180
<b>B8 Software Engineering</b>			<b>6 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
5. Sem.	Software Engineering	P	180
<b>B9 Informatik und Gesellschaft</b>			<b>5 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
6. Sem.	Informatik und Gesellschaft	P	150
<b>B10 Softwareprojekt (Lehramtsstudierende GyGe)</b>			<b>7 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
6. Sem	Softwareprojekt für Lehramtsstudierende – GyGe	P	210
<b>S1 Schlüsselqualifikation</b>			<b>4 LP</b>
<b>Zeitpunkt (Sem.)</b>		<b>P/WP</b>	<b>Workload (h)</b>
5. Sem	a) Proseminar Informatik	WP	120

4. § 42 wird wie folgt geändert:

a) Es wird folgender Absatz 3 eingefügt:

„(3) Die letzte Wiederholung einer Prüfung in Klausurform kann gemäß § 25 Absatz 4 Allgemeine Bestimmungen auf Wunsch der Kandidatin bzw. des Kandidaten als mündliche Ersatzprüfung durchgeführt werden. Die Dauer der mündlichen Ersatzprüfung beträgt 20 bis 45 Minuten.“

b) Der bisherige Absatz 3 wird zu Absatz 4 und wie folgt geändert:

„(4) Als Studienleistung kommt insbesondere in Betracht:

- Übungsaufgaben, die in der Regel wöchentlich als Hausaufgaben und/oder Präsenzaufgaben gestellt werden.
- Minitests: Zwei semesterbegleitende Kurztests, deren Dauer in der Regel nicht mehr als 30 Minuten beträgt.“

Näheres regeln die Modulbeschreibungen. Sofern in den Modulbeschreibungen Rahmenvorgaben enthalten sind, setzt die bzw. der jeweilige Lehrende fest, wie die Studienleistung konkret zu erbringen ist. Dies wird spätestens in den ersten drei Wochen der Vorlesungszeit von der bzw. dem jeweiligen Lehrenden und im Campus Management System der Universität Paderborn oder in sonstiger geeigneter Weise bekannt gegeben.

c) Der bisherige Absatz 4 wird zu Absatz 5.

5. Der Anhang „Studienverlaufsplan: Bachelor Lehramt GyGe Informatik“ wird wie folgt gefasst:

**Studienverlaufsplan: Bachelor Lehramt GyGe Informatik**

Semester	Modul/ Veranstaltung	Modul/ Veranstaltung	Σ LP
1	Programmierung I	Modellierung	16
2	Datenbanksysteme	Einführung in Computer Systeme	12
3	Datenstrukturen, Algorithmen und formale Sprachen	Didaktische Grundlagen des Informatikunterrichts – GyGe: a) Fachdidaktische Grundlagen	11
4	Programmierung II	Didaktische Grundlagen des Informatikunterrichts – GyGe: b) Stufenbezogene Unterrichtsmodelle – GyGe	11
5	Software Engineering	Schlüsselqualifikation Proseminar	10
6	Informatik und Gesellschaft	Softwareprojekt (Lehramtsstudierende GyGe) Softwareprojekt für Lehramtsstudierende – GyGe	12
		Summe:	72
		+ ggf. Bachelorarbeit 12 LP	

6. Der Anhang „Modulbeschreibungen“ wird wie folgt geändert:

**Modulbeschreibungen**

Programmierung I							
Programmierung I							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
B1	240	8	1.	WiSe	1	de	P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Programmierung I	V Ü	45 60	135	P	450 25	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> keine						
<b>4</b>	<p><b>Inhalte:</b></p> <p>Die Veranstaltung richtet sich an Studierende mit wenigen oder keinen Vorkenntnissen in der Softwareentwicklung. Sie führt die Studierenden an diese Tätigkeit heran, indem sie auf systematische und wissenschaftlich fundierte Art und Weise die grundlegenden Bausteine der imperativen und objektorientierten Programmierung in Python vermittelt und deren Anwendung praktisch einübt.</p> <p>Die Veranstaltung umfasst folgende Inhalte:</p> <ul style="list-style-type: none"> <li>• Grundlegende Programmierbausteine (Variablen, Operatoren und Ausdrücke, Zuweisungen, Verzweigungen, Schleifen, Ein- und Ausgabe)</li> <li>• Elementare Datentypen und Typverträglichkeit (Integer, String, Float, Boolean)</li> <li>• Funktionen (Definition, Aufruf, Parameter, Rückgabewerte)</li> <li>• Automatische Tests (Entwurf, Implementierung, Ausführung)</li> <li>• Datentypen (Arrays, Dictionaries)</li> <li>• Rekursion (rekursive Problemreduktion, Abbruchbedingung, Zusammenhang zur Iteration)</li> <li>• Klassen (Klassen und Objekte, Attribute und Methoden, Überladung, Sichtbarkeit, Delegation, Gleichheit und Identität, statische und nicht-statische Attribute)</li> <li>• Ausnahmen (Definition, Behandlung)</li> <li>• Dateizugriff und Betriebssystemaufrufe (Lesen und Schreiben, sequentiell und Direktzugriff)</li> </ul>						
<b>5</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b></p> <p>Nach Abschluss dieses Moduls können Studierende</p> <ul style="list-style-type: none"> <li>• Syntax und Semantik der wesentlichen Sprachkonstrukte einer objektorientierten Sprache erläutern</li> <li>• Grundkonzepte der imperativen und objektorientierten Programmierung erläutern, einfache algorithmische Lösungsmuster nachvollziehen und diese passend für gegebene Problemstellungen auswählen</li> <li>• einfache Algorithmen und Objektstrukturen zur Problemlösung entwerfen, implementieren und testen</li> <li>• für einfache Anwendungsprobleme eine geeignete programmtechnische Lösung auswählen</li> <li>• grundlegende Programmwurfprinzipien und -methoden anwenden</li> <li>• der Anwendungsdomäne entsprechende einfache Objektstrukturen entwerfen</li> <li>• die grundlegenden Werkzeuge der Softwareentwicklung effektiv einsetzen</li> </ul>						

	<ul style="list-style-type: none"> <li>• Lösungsansätze für Programmierprobleme gemeinsam in Kleingruppen entwickeln</li> <li>• Lösungsansätze für Programmierprobleme bewerten</li> <li>• eigene analytische und konzeptionelle Fähigkeiten einschätzen</li> </ul>								
6	<p><b>Prüfungsleistung:</b>  <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP)      <input type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1"> <thead> <tr> <th>zu</th> <th>Prüfungsform</th> <th>Dauer bzw. Umfang</th> <th>Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td></td> <td>Klausur</td> <td>120-180 Minuten</td> <td>100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Klausur	120-180 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Klausur	120-180 Minuten	100 %						
7	<p><b>Studienleistung / qualifizierte Teilnahme:</b></p> <table border="1"> <thead> <tr> <th>zu</th> <th>Form</th> <th>Dauer bzw. Umfang</th> <th>SL / QT</th> </tr> </thead> <tbody> <tr> <td></td> <td>Übungsaufgaben oder Minitests</td> <td></td> <td>SL</td> </tr> </tbody> </table> <p>Von der bzw. dem jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT		Übungsaufgaben oder Minitests		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
	Übungsaufgaben oder Minitests		SL						
8	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b>                  Bestandene Studienleistung</p>								
9	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b>                  Bestandene Modulabschlussprüfung</p>								
10	<p><b>Gewichtung für Gesamtnote:</b>                  Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>								
11	<p><b>Verwendung des Moduls in anderen Studiengängen:</b>                  Dieses Modul findet auch Verwendung in den Studiengängen B.Sc. Informatik, B.Sc. Computer Engineering, B.Ed. HRSGe Informatik sowie im Studiengang B.Ed. BK Informatik.</p>								
12	<p><b>Modulbeauftragte/r:</b>                  Prof. Dr. Stefan Böttcher</p>								
13	<p><b>Sonstige Hinweise:</b></p> <p><b>Methodische Umsetzung</b>                  Die Inhalte werden im Vorlesungsteil an typischen Beispielen eingeführt, in den praktischen Übungen unter Anleitung erprobt und in Übungsaufgaben in eigenständiger Bearbeitung der Studierenden vertieft. Dabei wird die Anwendung der wichtigsten Softwareentwicklungswerkzeuge wie Editor, Interpreter und Debugger eingeübt.</p> <p><b>Literaturangaben</b></p> <ul style="list-style-type: none"> <li>• Mark Lutz: Learning Python, 5th ed., O'Reilly</li> <li>• Thomas Theis: Einstieg in Python, Rheinwerk Computing</li> </ul> <p>Weitere Literatur wird in der Veranstaltung bekanntgegeben.</p>								

Modellierung							
Modelling							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
B2	240	8	1.	WiSe	1	de	P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Modellierung	V Ü	60 30	150	P	450 25	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> keine						
<b>4</b>	<p><b>Inhalte:</b></p> <p>Das Modellieren ist eine für das Fach Informatik typische Arbeitsmethode, die in allen Gebieten des Faches angewandt wird. Aufgaben, Probleme oder Strukturen werden untersucht und als Ganzes oder in Teilaspekten beschrieben, bevor sie durch den Entwurf von Software, Algorithmen, Daten und/oder Hardware gelöst bzw. implementiert werden. Mit der Modellierung eines Problems zeigt man, ob und wie es verstanden wurde. Damit ist sie Voraussetzung und Maßstab für die Lösung und sie liefert meist auch den Schlüssel für einen systematischen Entwurf. Als Ausdrucksmittel für die Modellierung steht ein breites Spektrum von Kalkülen und Notationen zur Verfügung. Sie sind spezifisch für unterschiedliche Arten von Problemen und Aufgaben. Deshalb werden in den verschiedenen Gebieten der Informatik unterschiedliche Modellierungsmethoden eingesetzt. In den entwurfsorientierten Gebieten (Softwaretechnik, Hardware-Entwurf) ist die Bedeutung der Modellierung und die Vielfalt der Methoden natürlich besonders stark ausgeprägt.</p> <ul style="list-style-type: none"> <li>• Grundlegende Kalküle: Wertebereiche, Terme, Algebren</li> <li>• Logik: Aussagenlogik, Prädikatenlogik</li> <li>• Modellierung mit Graphen: Weg, Verbindung, Zuordnung, Abhängigkeiten, Abfolgen</li> <li>• Grammatiken: reguläre und kontextfreie Grammatiken</li> <li>• Modellierung von Abläufen: endliche Automaten</li> <li>• Modellierung von kontinuierlichen Prozessen und Funktionen</li> </ul>						
<b>5</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b></p> <p>Die Studierenden können</p> <ul style="list-style-type: none"> <li>• grundlegende Literatur im Bereich der Modellierung verstehen,</li> <li>• für ein gegebenes Problem eine geeignete Modellierungstechnik auswählen und das Problem mit dieser Technik beschreiben</li> </ul> <p>Der erfolgreiche Abschluss dieses Moduls ermächtigt Studierende, erlernte Modellierungstechniken anzuwenden, grundsätzliche Techniken zu erweitern und verfeinern, um somit neuartige Probleme zu modellieren. Sie werden darüber hinaus ermächtigt, diese Techniken im Rahmen verschiedener Anwendungen zu implementieren, analysieren und bewerten.</p>						

6	<b>Prüfungsleistung:</b> [X] Modulabschlussprüfung (MAP)      [ ] Modulprüfung (MP)      [ ] Moduleilprüfungen (MTP)			
	<b>zu</b>	<b>Prüfungsform</b>	<b>Dauer bzw. Umfang</b>	<b>Gewichtung für die Modulnote</b>
		Klausur	120-180 Minuten	100 %
7	<b>Studienleistung / qualifizierte Teilnahme:</b>			
	<b>zu</b>	<b>Form</b>	<b>Dauer bzw. Umfang</b>	<b>SL / QT</b>
		Übungsaufgaben		SL
Von der bzw. dem jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.				
8	<b>Voraussetzungen für die Teilnahme an Prüfungen:</b> Bestandene Studienleistung			
9	<b>Voraussetzungen für die Vergabe von Leistungspunkten:</b> Bestandene Modulabschlussprüfung			
10	<b>Gewichtung für Gesamtnote:</b> Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).			
11	<b>Verwendung des Moduls in anderen Studiengängen:</b> Dieses Modul findet auch Verwendung im Studiengang B.Sc. Informatik sowie im Studiengang B.Ed. BK Informatik.			
12	<b>Modulbeauftragte/r:</b> Prof. Dr. Christian Scheideler, Prof. Dr. Johannes Blömer, Prof. Dr. Sevag Gharibian			
13	<b>Sonstige Hinweise:</b>			
	<b>Methodische Umsetzung</b> In der Vorlesung werden <ul style="list-style-type: none"> <li>• die grundlegenden Begriffe eingeführt,</li> <li>• mathematische Konzepte formuliert, analysiert und in Teilen bewiesen</li> <li>• die theoretischen Inhalte anhand einfacher Beispiele veranschaulicht</li> </ul> In den Übungen werden <ul style="list-style-type: none"> <li>• die erlernten Konzepte umfassend auf Fragestellungen verschiedener Komplexität angewendet</li> <li>• weiterführende Beispiele betrachtet, um das Erlernete zu erweitern bzw. um weitere Perspektiven zu ergänzen</li> </ul> Die Studienleistung <ul style="list-style-type: none"> <li>• dient der besseren Vorbereitung auf die Abschlussprüfung</li> <li>• motiviert Studierende, kontinuierlich mitzuarbeiten</li> <li>• ermöglicht regelmäßige Rückmeldungen über den eigenen Lernfortschritt</li> </ul> <b>Lernmaterialien, Literaturangaben</b> <ul style="list-style-type: none"> <li>• Uwe Kastens, Hans Kleine Büning: Modellierung</li> <li>• Angelika Steger: Diskrete Strukturen</li> <li>• Foliensatz der Vorlesung; Übungsblätter</li> </ul>			



Datenbanksysteme							
Database Systems							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
B3	180	6	2.	SoSe	1	de	P
1	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Datenbanksysteme	V Ü	45 30	105	P	400 25	
2	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
3	<b>Teilnahmevoraussetzungen:</b> <b>Empfohlene Vorkenntnisse:</b> Kenntnisse in der Programmierung werden in dem Umfang empfohlen, wie sie im Modul Programmierung I gelehrt werden. Elementare Kenntnisse der Logik und der Modellierung aus dem Modul Modellierung werden empfohlen.						
4	<p><b>Inhalte:</b></p> <p>Datenbanken spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissens der Unternehmen als Daten in Datenbanken gespeichert wird. Für das Unternehmen ist es entscheidend, dass diese Daten korrekt, insbesondere konsistent, sind und dass sie effizient erfragt und aktualisiert werden können. Weiterhin sind die in Datenbanken abgelegten Datenbestände die wesentliche Datenquelle für eine Vielzahl von Anwendungsprogrammen, sie werden aber auch durch Anwendungsprogramme aktualisiert. Deshalb kommen der Organisation und Verarbeitung großer Datenbestände sowie der Einbindung von Datenbanken in Anwendungen eine zentrale Rolle bei der Erstellung korrekter und effizienter Anwendungen zu.</p> <p>Durch die Nutzung zunehmend größerer Datenbestände (Big Data) entstehen neuartige Anforderungen (Umgang mit Volume, Variety, Velocity), die neue Techniken in der Datenbankwelt erfordern. So werden verteilte Datenbanken immer wichtiger, und der Umgang mit Netzwerkunterbrechungen erfordert geschickte, je nach Anwendung unterschiedliche Kompromisse zwischen Erreichbarkeit, Aktualität und Konsistenz verteilter Daten. NoSQL-Datenbanken sind für diese speziellen Bedürfnisse konzipiert und setzen – im Gegensatz zu den traditionellen relationalen Datenbanksystemen – den Fokus auf eine deutlich flexiblere Datenorganisation.</p> <p>Dieses Modul erschließt die Grundlagen für verschiedenartige Datenbanksysteme, die in nahezu allen Unternehmen in der Praxis eingesetzt werden. Die Inhalte umfassen im Einzelnen:</p> <ul style="list-style-type: none"> <li>• Relationales Datenmodell und relationale Algebra</li> <li>• SQL – Datendefinitionssprache, Datenmanipulationssprache und Anfragesprache; Stored Procedures; Assertions</li> <li>• Zugriffskontrolle und Sichten – Views in SQL; Rechtevergabe in SQL</li> <li>• Physische Datenorganisation und Indizes</li> <li>• Anfrageoptimierung</li> <li>• Datenintegrität</li> <li>• Funktionale Abhängigkeiten und Datenbankschemaentwurf – Normalformen; Transformationseigenschaften; Integritätsbedingungen (Schlüssel und Fremdschlüssel); Trigger in SQL</li> <li>• Transaktionen – Synchronisation; Recovery; Atomic Commit Protokolle</li> <li>• NoSQL-Datenbanken – Graph-Datenbanken; Dokumenten-orientierte Datenbanken; Key-Value Stores; Column Stores; Frameworks zur Auswertung verteilter Daten/Streamprocessing (Map/Reduce, Apache Spark, Apache Flink)</li> <li>• Hauptspeicher-Datenbanken</li> <li>• Eingebettetes SQL – SQL-Einbettung in Java bzw. Python; SQL Injection und Prepared Statements</li> <li>• Datenschutz in Datenbanken – Herausforderungen und Grenzen</li> </ul>						

5	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b>                  Nach Abschluss dieses Moduls können Studierende</p> <ul style="list-style-type: none"> <li>• Operatoren der relationalen Anfragesprachen benennen und deren Bedeutungen erklären</li> <li>• Anfragen in relationaler Algebra, SQL und Cypher korrekt interpretieren und formulieren und an existierende relationale Datenbanken bzw. Graph-Datenbanken stellen</li> <li>• Programme schreiben, die Datenbestände aus Datenbanken lesen oder verändern</li> <li>• ein Datenbankschema möglichst redundanzfrei entwerfen und darauf aufbauend eine Datenbank definieren und aufbauen. Weiterhin können die Studierenden die Risiken eines schlecht entworfenen Datenbankschemas bewerten</li> <li>• verteilte Datenanalyse mit Map/Reduce, Apache Spark und Apache Flink durchführen</li> <li>• den Programmieraufwand für Datenbankabfragen und Datenbankprogrammierung einschätzen sowie Anfragen manuell optimieren</li> <li>• die Folgen einer Datenbankschema-Änderung erkennen und abschätzen</li> <li>• den Aufwand und Nutzen von Synchronisation und Recovery verstehen und die Techniken und Verfahren in diesen Bereichen praktisch einsetzen</li> <li>• die Eignung und Grenzen des relationalen Datenmodells bewerten und einschätzen und mit alternativen Datenmodellen vergleichen</li> <li>• die erworbenen Kompetenzen und Fertigkeiten auf andere Datenquellen oder andere Datenbanksysteme übertragen</li> <li>• mit anderen Studierenden in Kleingruppen kooperieren, um gemeinsam eine geeignete Lösung für eine gegebene Aufgabe im Bereich der vermittelten Inhalte zu erstellen</li> </ul>								
6	<p><b>Prüfungsleistung:</b>  <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP)      <input type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" data-bbox="256 1099 1477 1223"> <thead> <tr> <th>zu</th> <th>Prüfungsform</th> <th>Dauer bzw. Umfang</th> <th>Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td></td> <td>Klausur</td> <td>90-120 Minuten</td> <td>100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Klausur	90-120 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Klausur	90-120 Minuten	100 %						
7	<p><b>Studienleistung / qualifizierte Teilnahme:</b></p> <table border="1" data-bbox="264 1261 1481 1402"> <thead> <tr> <th>zu</th> <th>Form</th> <th>Dauer bzw. Umfang</th> <th>SL / QT</th> </tr> </thead> <tbody> <tr> <td></td> <td>Übungsaufgaben</td> <td></td> <td>SL</td> </tr> </tbody> </table> <p>Von der bzw. dem jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT		Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
	Übungsaufgaben		SL						
8	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b>                  Bestandene Studienleistung</p>								
9	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b>                  Bestandene Modulabschlussprüfung</p>								
10	<p><b>Gewichtung für Gesamtnote:</b>                  Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>								
11	<p><b>Verwendung des Moduls in anderen Studiengängen:</b>                  Das Modul findet auch Verwendung im Studiengang B.Sc. Informatik, im Studiengang B. Ed. HRSGe sowie im Studiengang B.Ed. BK Informatik.</p>								
12	<p><b>Modulbeauftragte/r:</b>                  Dr. Rita Hartel, Prof. Dr. Stefan Böttcher</p>								
13	<p><b>Sonstige Hinweise:</b>  <b>Methodische Umsetzung</b></p>								

<p>Die Grundlagen und Konzepte von Datenbanksystemen werden in einer Präsentation im Rahmen einer Vorlesung eingeführt und anschließend in Präsenzübungen in Kleingruppen sowie in Übungsaufgaben vertieft und durch praktische Übungen ergänzt.</p> <p><b>Lernmaterialien, Literaturangaben</b></p> <ul style="list-style-type: none"> <li>Lehrbuch: Kemper, Eickler: Datenbanksysteme. Eine Einführung. Oldenbourg-Verlag, neueste Ausgabe.</li> <li>Lehrbuch: Garcia-Molina, Ullman, Widom: Database Systems: The Complete Book, Prentice Hall, neueste Ausgabe.</li> <li>Lehrbuch: Heuer, Saake: Datenbanksysteme – Konzepte und Sprachen. Mitp-Verlag, neueste Ausgabe.</li> <li>Lehrbuch: Sadalage, Fowler: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, neueste Ausgabe.</li> </ul>
--

Einführung in Computer Systeme							
Introduction to Computer Systems							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
B4	180	6	2.	SoSe	1	de	P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Einführung in Computer Systeme	V Ü	30 30	120	P	50 25	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> keine						
<b>4</b>	<b>Inhalte:</b> Die Veranstaltung gibt einen Überblick über Computer Systeme und umfasst unter anderem folgende Inhalte: <ul style="list-style-type: none"> <li>• Darstellung von Information durch Daten,</li> <li>• Codierungen</li> <li>• Aufbau und Funktionsweise von Rechnern und Rechnernetzen, Grundlagen von Informations- und Kommunikationssystemen</li> <li>• Grundlagen, Merkmale, Einsatzgebiete und Verwendung von Betriebssystemen</li> <li>• Sicherheit, Sicherheit von IT-Systemen</li> <li>• Grundlagen von Schaltkreisen, Grundlagen der Elektrotechnik mit Bezug zur Technischen Informatik</li> <li>• Netzstrukturen, verteilte Systeme und Basistechnologien</li> <li>• Protokollarchitektur</li> <li>• Grundlagen der Kryptographie</li> </ul>						

5	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b></p> <p><b>Fachliche Kompetenzen:</b>                  Absolventen der Lehrveranstaltung können die Darstellung von Information durch Datenbeschreiben und den Aufbau und die Funktionsweise von Rechnern und Rechnernetzen benennen, sie verstehen und analysieren. Die Studierenden lernen Grundlagen zu Methoden der Codierung. Im Rahmen der Veranstaltung werden grundlegende Konzepte von Betriebssystemen besprochen. Ein weiteres Thema der Veranstaltung ist Sicherheit.                  Es findet eine stärkere Vertiefung zu den Inhalten aus der Vorlesung statt, außerdem werden weitere Themen wie zum Beispiel die Grundlagen von Schaltkreisen besprochen.</p> <p><b>Schlüsselkompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> <li>• Gruppenarbeit</li> </ul>								
6	<p><b>Prüfungsleistung:</b>  <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP)      <input type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" data-bbox="256 779 1473 936"> <thead> <tr> <th data-bbox="256 779 376 853">zu</th> <th data-bbox="376 779 970 853">Prüfungsform</th> <th data-bbox="970 779 1230 853">Dauer bzw. Umfang</th> <th data-bbox="1230 779 1473 853">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td data-bbox="256 853 376 936"></td> <td data-bbox="376 853 970 936">Mündliche Prüfung oder Klausur</td> <td data-bbox="970 853 1230 936">ca. 30 Minuten 90-120 Minuten</td> <td data-bbox="1230 853 1473 936">100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Mündliche Prüfung oder Klausur	ca. 30 Minuten 90-120 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Mündliche Prüfung oder Klausur	ca. 30 Minuten 90-120 Minuten	100 %						
7	<p><b>Studienleistung / qualifizierte Teilnahme:</b>                  Qualifizierte Teilnahme zu der Lehrveranstaltung des Moduls gemäß § 42 Besondere Bestimmungen. Näheres zu Form und Umfang bzw. Dauer gibt die Lehrende bzw. der Lehrende spätestens in den ersten drei Wochen der Vorlesungszeit bekannt.</p>								
8	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b>                  keine</p>								
9	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b>                  Bestandene Modulabschlussprüfung sowie qualifizierte Teilnahme an der Veranstaltung des Moduls</p>								
10	<p><b>Gewichtung für Gesamtnote:</b>                  Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>								
11	<p><b>Verwendung des Moduls in anderen Studiengängen:</b>                  Dieses Modul findet auch Verwendung im Studiengang B.Ed. HRSGe Informatik sowie im Studiengang B.Ed. BK Informatik.</p>								
12	<p><b>Modulbeauftragte/r:</b>                  Prof. Dr. Carsten Schulte</p>								
13	<p><b>Sonstige Hinweise:</b>                  keine</p>								

Datenstrukturen, Algorithmen und formale Sprachen							
Data Structures, Algorithms and Formal Languages							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
B5	240	8	3.	WiSe	1	de	P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Datenstrukturen, Algorithmen und formale Sprachen	V Ü	60 30	150	P	50 25	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> <b>Empfohlene Vorkenntnisse</b> Kenntnisse in der Programmierung werden in dem Umfang empfohlen, wie sie in der Veranstaltung „Programmierung I“ gelehrt werden.						
<b>4</b>	<p><b>Inhalte:</b></p> <p>Algorithmen sind die Grundlage jeder Hard- und Software und spielen eine zentrale Rolle in der Informatik. Ein wichtiges Ziel des Algorithmenentwurfs ist die Effizienz, d. h. die Entwicklung von Algorithmen, die ein Problem möglichst schnell oder mit möglichst geringem Speicherbedarf lösen. Diese sind eng mit effizienten Datenstrukturen verknüpft, die große Datenmengen so organisieren, dass einfache Abfragen wie Suchen, komplexe Abfragen oder Änderungsoperationen wie Einfügen und Löschen effizient beantwortet werden können. Begleitend werden wichtige Algorithmen und Datenstrukturen im Rahmen dieser Vorlesung und der dazugehörigen Übungen selbst programmiert.</p> <p>Darüber hinaus werden grundlegende Methoden und Techniken zur Charakterisierung der Schwierigkeit von Berechnungsproblemen vorgestellt. Als formales Rechenmodell werden Turingmaschinen definiert. Darauf aufbauend werden die wichtigsten Konzepte und Techniken der Berechenbarkeitstheorie und Komplexitätstheorie (z.B. Zeitkomplexität, P- und NP-Klassen, NP-Vollständigkeit, polynomielle Reduktionen) erläutert.</p> <p><i>Inhaltsübersicht:</i></p> <ul style="list-style-type: none"> <li>• Mathematische Grundlagen mit Anwendungsbezug</li> <li>• Algorithmische Prinzipien: Rekursion, Teile-und-Herrsche, systematische Suche, Backtracking</li> <li>• Sortierverfahren: Quicksort, Heapsort, Mergesort</li> <li>• Datenstrukturen: Verkettete Listen, (balancierte) Bäume, Graphen, Hashing, Prioritätswarteschlangen</li> <li>• Graphenalgorithmen: Kürzeste Wege, Minimale Spannbäume</li> <li>• Kellerautomaten und Turing-Maschinen</li> <li>• Zeit- und Platzkomplexität von Algorithmen</li> <li>• Berechenbarkeit und ihre Grenzen</li> <li>• Berechenbarkeits- und Komplexitätsklassen</li> <li>• Asymptotisches Wachstum von Komplexität</li> <li>• NP-Vollständigkeit und Reduktionen</li> <li>• Spezifikation und Verifikation von Programmen</li> </ul>						

5	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b>                  Nach Abschluss des Moduls können die Studierenden</p> <ul style="list-style-type: none"> <li>• effiziente Datenstrukturen und Algorithmen für ausgewählte grundlegende Probleme erklären und anwenden,</li> <li>• Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen einsetzen,</li> <li>• selbstständig und kreativ Algorithmen und Datenstrukturen entwickeln,</li> <li>• mathematische Methoden zum Korrektheitsbeweis und zur Effizienzanalyse einsetzen,</li> <li>• die Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten einschätzen,</li> <li>• sich neue Algorithmen, Datenstrukturen und algorithmische Ideen und Analysen aneignen,</li> <li>• selbstständig im Bereich der Berechenbarkeit und Komplexitätstheorie analysieren und klassifizieren,</li> <li>• im Rahmen einer formalen Spezifikation einfacher Programme festlegen, was diese leisten sollen,</li> <li>• verifizieren, dass ein einfaches Programm entsprechend einer gegebenen Spezifikation korrekt ist.</li> </ul>								
6	<p><b>Prüfungsleistung:</b>  <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP)      <input type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" data-bbox="255 784 1476 940"> <thead> <tr> <th>zu</th> <th>Prüfungsform</th> <th>Dauer bzw. Umfang</th> <th>Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td></td> <td>Mündliche Prüfung oder Klausur</td> <td>ca. 30 Minuten 90-120 Minuten</td> <td>100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Mündliche Prüfung oder Klausur	ca. 30 Minuten 90-120 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Mündliche Prüfung oder Klausur	ca. 30 Minuten 90-120 Minuten	100 %						
7	<p><b>Studienleistung / qualifizierte Teilnahme:</b></p> <table border="1" data-bbox="263 985 1476 1097"> <thead> <tr> <th>zu</th> <th>Form</th> <th>Dauer bzw. Umfang</th> <th>SL / QT</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Übungsaufgaben und/oder Minitests</td> <td></td> <td>SL</td> </tr> </tbody> </table> <p>Von der bzw. dem jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Übungsaufgaben und/oder Minitests		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Übungsaufgaben und/oder Minitests		SL						
8	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b>                  Bestandene Studienleistung</p>								
9	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b>                  Bestandene Modulabschlussprüfung</p>								
10	<p><b>Gewichtung für Gesamtnote:</b>                  Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>								
11	<p><b>Verwendung des Moduls in anderen Studiengängen:</b>                  Dieses Modul findet auch Verwendung im Studiengang B.Ed. BK Informatik.</p>								
12	<p><b>Modulbeauftragte/r:</b>                  Dr. Harald Selke</p>								
13	<p><b>Sonstige Hinweise:</b></p> <p><b>Methodische Umsetzung</b>                  Die Inhalte werden in einer Präsentation im Rahmen einer Vorlesung eingeführt und anschließend in Präsenzübungen in Kleingruppen vertieft und durch praktische Übungen ergänzt.</p> <p><b>Lernmaterialien, Literaturangaben</b></p> <ul style="list-style-type: none"> <li>• Thomas H. Cormen, Algorithmen – Eine Einführung</li> <li>• Michael Sipser: Introduction to the Theory of Computation</li> <li>• Uwe Schöning: Theoretische Informatik – kurz gefasst</li> </ul>								

Didaktische Grundlagen des Informatikunterrichts – GyGe							
Foundations of Computer Science Education – GyGe							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache: de	P/WP: P
B6	240	8	3. und 4.	WiSe / SoSe	2		
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	a) Fachdidaktische Grundlagen	V	30	60	P	50	
	b) Stufenbezogene Unterrichtsmodelle – GyGe	S	60	90	P	30	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> keine						
<b>4</b>	<b>Inhalte:</b> Zu a) Die Veranstaltung gibt einen Überblick über didaktische Ansätze und umfasst unter anderem folgende Inhalte: <ul style="list-style-type: none"> <li>• Grundfragen der Informatikdidaktik</li> <li>• Informatikspezifische Medien und Methoden</li> <li>• Informatische Bildungskonzepte</li> <li>• Grundlagen der fachspezifischen Diagnostik im Informatikunterricht</li> <li>• Informatikunterricht und Lerntheorien</li> </ul> Zu b) Die Veranstaltung gibt einen Überblick über die Planung, Organisation und Durchführung von Informatikunterricht und umfasst unter anderem folgende Inhalte: <ul style="list-style-type: none"> <li>• Planung, Durchführung und Analyse von stufenbezogenen Unterrichtseinheiten in Gymnasien und Gesamtschulen</li> <li>• Analyse und Reflexion von Lehr- und Lernprozessen</li> <li>• Einsatz von Unterrichtssoftware und Lernumgebungen im Informatikunterricht in Gymnasien und Gesamtschulen</li> <li>• Handlungsorientierter Informatikunterricht (z.B. Plan- und Rollenspiele, Informatik unplugged)</li> <li>• Umgang mit Heterogenität (u. a. Leistungsdifferenzierung, Genderaspekte, Förderung)</li> </ul>						
<b>5</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen:</b> <b>Fachliche Kompetenzen:</b> Die Studierenden sind in der Lage, Grundsätze und Standards für den Informatikunterricht zu nennen und auf konkrete Lernsituationen zu beziehen. Dazu können sie Informatikunterricht theoriebasiert aufbereiten (Rekonstruktion fachlichen Wissens) und unter Berücksichtigung von individuellen Voraussetzungen und Fähigkeiten der Schülerinnen und Schüler Lehr- und Lernprozesse im Informatikunterricht analysieren, durchführen und bewerten. Zu a) Die Studierenden sind in der Lage, didaktische (Re-)Konstruktionen von fachlichem Wissen durchzuführen und insbesondere didaktische Reduktionen zu analysieren und zu bewerten. Methoden, Techniken und Medien zur Erschließung informatischer Inhalte werden analysiert, so dass die visuelle, auditive und haptische Wahrnehmung angesprochen wird.						

	<p>Zu b) Die Studierenden sind in der Lage, Lehr- und Lernprozesse im Informatikunterricht unter Berücksichtigung der individuellen Voraussetzungen und Fähigkeiten aller Schülerinnen und Schüler durchzuführen, zu analysieren und zu bewerten.</p> <p><b>Schlüsselkompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Lernmotivation</li> </ul>		
<b>6</b>	<p><b>Prüfungsleistung:</b>  <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP)      <input type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p>		
	<b>zu</b>	<b>Prüfungsform</b>	<b>Dauer bzw. Umfang</b>
	a) und b)	Portfolio	50.000-62.500 Zeichen
	<b>Gewichtung für die Modulnote</b>	100 %	
<b>7</b>	<p><b>Studienleistung / qualifizierte Teilnahme:</b>                  Qualifizierte Teilnahme zu den Lehrveranstaltungen des Moduls gemäß § 42 Besondere Bestimmungen. Näheres zu Form und Umfang bzw. Dauer gibt die Lehrende bzw. der Lehrende spätestens in den ersten drei Wochen der Vorlesungszeit bekannt.</p>		
<b>8</b>	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b>                  Keine</p>		
<b>9</b>	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b>                  Bestandene Modulabschlussprüfung sowie qualifizierte Teilnahme an den Veranstaltungen des Moduls</p>		
<b>10</b>	<p><b>Gewichtung für Gesamtnote:</b>                  Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>		
<b>11</b>	<p><b>Verwendung des Moduls in anderen Studiengängen:</b>                  keine</p>		
<b>12</b>	<p><b>Modulbeauftragte/r:</b>                  Prof. Dr. Carsten Schulte</p>		
<b>13</b>	<p><b>Sonstige Hinweise:</b>                  Dieses Modul beinhaltet die Auseinandersetzung mit inklusionsrelevanten Fragestellungen im Umfang eines Workloads von 2 LP.</p>		



Programmierung II							
Programming II							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
B7	180	6	4.	SoSe	1	de	P
1	<b>Modulstruktur:</b>						
		Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
		Programmierung II	V Ü	45 30	105	P	50 25
2	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
3	<b>Teilnahmevoraussetzungen:</b> <b>Empfohlene Vorkenntnisse</b> Kenntnisse in der Programmierung werden in dem Umfang empfohlen, wie sie in der Veranstaltung „Programmierung I“ gelehrt werden.						
4	<p><b>Inhalte:</b></p> <p>Die Veranstaltung setzt auf den Kenntnissen der Lehrveranstaltung Programmierung I auf und erweitert und vertieft diese. Im Bereich der Programmierung liegt der Schwerpunkt auf objektorientierter Programmierung sowie der Konzeption und Implementierung von grundlegenden Algorithmen und Datenstrukturen wie Such- und Sortierverfahren, Bäumen und Graphen und ihrer Traversierung sowie auf der korrekten Implementierung nebenläufiger Programme. Programmiersprachen und -paradigmen, die von der in Programmierung I betrachteten imperativen Programmierung abweichen, werden im Rahmen dieser Lehrveranstaltung vermittelt.</p> <p>Weiterhin führt diese Lehrveranstaltung in die Grundlagen des maschinellen Lernens ein. Es werden sowohl grundlegende überwachte und unüberwachte Algorithmen des maschinellen Lernens als auch komplexere Deep-Learning-Ansätze, wie generative Modelle, vorgestellt.</p> <p>Die Veranstaltung umfasst folgende Inhalte:</p> <ul style="list-style-type: none"> <li>• Vererbung (Ober- und Unterklassen, Überschreiben und Überdecken, abstrakte Methoden und Klassen, multiple Vererbung)</li> <li>• Bibliotheken (Einbindung, Anwendung)</li> <li>• Array-basierte Such- und Sortieralgorithmen (binäre Suche, Rank, Bubble-, Index- und Radix-Sort)</li> <li>• Dynamische Datenstrukturen (lineare Liste, Ringliste, Binär- und allgemeine Bäume)</li> <li>• Elementare Algorithmen auf Graphen (Implementierung von Graphen durch Arrays, Tiefen- und Breiten-suche, Zyklensuche)</li> <li>• Nebenläufige Programmierung</li> <li>• Einblick in die funktionale und logische Programmierung</li> <li>• Überwachte Klassifizierung/Regression</li> <li>• Unüberwachtes Repräsentationslernen</li> <li>• Verstärkendes Lernen</li> <li>• Modelauswahl und Evaluation</li> </ul>						
5	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b></p> <p>Nach Abschluss des Moduls können die Studierenden</p> <ul style="list-style-type: none"> <li>• fortgeschrittene Konzepte objektorientierter Programmiersprachen erklären und anwenden</li> <li>• die grundlegenden Such- und Sortieralgorithmen erklären, implementieren und anwenden</li> </ul>						

	<ul style="list-style-type: none"> <li>• dynamische Datenstrukturen verstehen, sinnvoll einsetzen und implementieren</li> <li>• Datenstrukturen und Algorithmen aus Softwarebibliotheken finden und nutzen</li> <li>• selbständig dynamische Datenstrukturen entwerfen und implementieren</li> <li>• selbständig vollständige Programme kleinen und mittleren Umfangs erstellen</li> <li>• einfache nebenläufige Programme erstellen</li> <li>• die grundlegenden syntaktischen und semantischen Konzepte und typischen Eigenschaften von Logischer und Funktionaler Programmierung erklären</li> <li>• die Stärken, Schwächen und Herausforderungen der verschiedenen Programmierparadigmen erläutern und ein geeignetes Programmierparadigma für ein gegebenes Problem auswählen.</li> <li>• verschiedene Modelle des maschinellen Lernens erläutern</li> <li>• diese Modelle selbstständig evaluieren</li> <li>• maschinelle Lernmodelle trainieren</li> </ul>								
<b>6</b>	<p><b>Prüfungsleistung:</b>  <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP)      <input type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1"> <thead> <tr> <th>zu</th> <th>Prüfungsform</th> <th>Dauer bzw. Umfang</th> <th>Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td></td> <td>Mündliche Prüfung oder Klausur</td> <td>ca. 30 Minuten 90 Minuten</td> <td>100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Mündliche Prüfung oder Klausur	ca. 30 Minuten 90 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Mündliche Prüfung oder Klausur	ca. 30 Minuten 90 Minuten	100 %						
<b>7</b>	<p><b>Studienleistung / qualifizierte Teilnahme:</b></p> <table border="1"> <thead> <tr> <th>zu</th> <th>Form</th> <th>Dauer bzw. Umfang</th> <th>SL / QT</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Übungsaufgaben und/oder Minitests</td> <td></td> <td>SL</td> </tr> </tbody> </table> <p>Von der bzw. dem jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Übungsaufgaben und/oder Minitests		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Übungsaufgaben und/oder Minitests		SL						
<b>8</b>	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b> Bestandene Studienleistung</p>								
<b>9</b>	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b> Bestandene Modulabschlussprüfung</p>								
<b>10</b>	<p><b>Gewichtung für Gesamtnote:</b> Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>								
<b>11</b>	<p><b>Verwendung des Moduls in anderen Studiengängen:</b> Dieses Modul findet auch Verwendung im Studiengang B.Ed. HRSGe Informatik sowie im Studiengang B.Ed. BK Informatik.</p>								
<b>12</b>	<p><b>Modulbeauftragte/r:</b> Prof. Dr. Carsten Schulte</p>								
<b>13</b>	<p><b>Sonstige Hinweise:</b> keine</p>								

Software Engineering							
Software Engineering							
<b>Modulnummer:</b> B8	<b>Workload (h):</b> 180	<b>LP:</b> 6	<b>Studiensemester:</b> 5.	<b>Turnus:</b> WiSe	<b>Dauer (in Sem.):</b> 1	<b>Sprache:</b> de	<b>P/WP:</b> P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Software Engineering	V Ü	30 30	120	P	300 25	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> Keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> Empfohlene Vorkenntnisse Kenntnisse aus den Modulen Programmierung I und Modellierung werden empfohlen.						
<b>4</b>	<p><b>Inhalte:</b></p> <p>In der Veranstaltung werden die Grundlagen der systematischen und ingenieurmäßigen Softwareentwicklung vermittelt. Im Fokus steht dabei eine durchgängige modellbasierte Entwicklungsmethode. Der Kurs führt in wesentliche Vorgehensmodelle für die Softwareentwicklung ein. Es werden Methoden für die Softwareentwicklung und Qualitätssicherung vermittelt, die innerhalb der Vorgehensmodelle zum Einsatz kommen. Außerdem werden Modellierungssprachen vorgestellt, mit denen die statischen und dynamischen Aspekte von Softwaresystemen beschrieben werden können. Hierzu wird die objektorientierte Modellierungssprache UML (Unified Modeling Language) eingeführt, die dedizierte Diagrammsprachen vereint. Methodische Hinweise zum Einsatz der Modellierungssprachen im Softwareentwicklungsprozess, zur Erstellung der Ergebnisartefakte und zur Prüfung ihrer Qualität werden gegeben. Modellierungswerkzeuge werden exemplarisch eingesetzt. Schließlich werden spezifische Programmierkonzepte und -werkzeuge für die Softwareentwicklung im Team betrachtet.</p> <p>Die Veranstaltung umfasst folgende Inhalte:</p> <ul style="list-style-type: none"> <li>• Einführung in die Konzepte und Begrifflichkeiten der Softwaretechnik (Software Engineering)</li> <li>• Softwarelebenszyklus und Vorgehensmodelle (klassische, agile, hybride)</li> <li>• Modellierungssprachen und modellbasierte Softwareentwicklung</li> <li>• UML (Unified Modeling Language): Klassendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme, Zustandsdiagramme, Komponentendiagramme</li> <li>• durchgängige Softwareentwicklungsmethode von der Anforderungsspezifikation über Modellierung (Softwarearchitektur, Softwareentwurf) bis zur Implementierung und zum Test der Software</li> <li>• Transformation und Konsistenz von Modellen, Codegenerierung</li> <li>• spezifische Programmierkonzepte (Objektorientierung, Typisierung, Schnittstellen) für die arbeitsteilige Softwareentwicklung im Team</li> <li>• Softwaretesten und modellbasiertes Testen</li> </ul> <p>Konzepte und Werkzeuge für Quellcode-Verwaltung, Continuous Integration/Delivery</p>						
<b>5</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b></p> <p>Die Studierenden können nach Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• für eine gegebene Problemstellung systematisch und schrittweise eine Softwarelösung unter Einhaltung eines Prozessmodells und Einsatz einer (modellbasierten) Entwicklungsmethode entwickeln</li> </ul>						

	<ul style="list-style-type: none"> <li>die Unterschiede von Vorgehensmodellen erklären, die Beziehungen und Übergänge zwischen verschiedenen Entwicklungsphasen eines Vorgehensmodells verstehen und für ein Entwicklungsprojekt ein geeignetes Vorgehen wählen</li> <li>wichtige Modellierungssprachen, insbesondere die verschiedenen Diagrammsprachen der UML, für die einzelnen Entwicklungsschritte bei der Modellierung der unterschiedlichen Aspekte einer Softwarelösung auswählen und verwenden, um korrekte und passende Softwaremodelle zu erstellen</li> <li>die Qualität von Softwaremodellen und (Zwischen-) Ergebnissen bewerten</li> <li>zur Überprüfung der Qualität der entwickelten Softwarelösung Techniken des Softwaretestens und des modellbasierten Testens einsetzen</li> <li>Konzepte und Softwareentwicklungswerkzeuge für die arbeitsteilige Softwareentwicklung im Team auswählen und einsetzen</li> </ul> <p>eine größere, zusammenhängende Softwaremodellierungsaufgabe über einen längeren Zeitraum selbstorganisiert im Team bearbeiten und hierbei die Bedeutung der verschiedenen Phasen, Methoden und Softwarewerkzeuge verstehen</p>								
<b>6</b>	<p><b>Prüfungsleistung:</b>  <input type="checkbox"/> Modulabschlussprüfung (MAP)      <input checked="" type="checkbox"/> Modulprüfung (MP)      <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1"> <thead> <tr> <th>zu</th> <th>Prüfungsform</th> <th>Dauer bzw. Umfang</th> <th>Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td></td> <td>Klausur</td> <td>90-120 Minuten</td> <td>100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Klausur	90-120 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Klausur	90-120 Minuten	100 %						
<b>7</b>	<p><b>Studienleistung / qualifizierte Teilnahme:</b> keine</p>								
<b>8</b>	<p><b>Voraussetzungen für die Teilnahme an Prüfungen:</b> keine</p>								
<b>9</b>	<p><b>Voraussetzungen für die Vergabe von Leistungspunkten:</b> Bestandene Modulprüfung</p>								
<b>10</b>	<p><b>Gewichtung für Gesamtnote:</b> Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</p>								
<b>11</b>	<p><b>Verwendung des Moduls in anderen Studiengängen:</b> Dieses Modul findet auch Verwendung im Studiengang B.Sc. Informatik sowie im Studiengang B.Ed. BK Informatik.</p>								
<b>12</b>	<p><b>Modulbeauftragte/r:</b> Prof. Dr. Stefan Sauer, Prof. Dr. Eric Bodden</p>								
<b>13</b>	<p><b>Sonstige Hinweise:</b>  <b>Methodische Umsetzung</b>                  In der Vorlesung werden die Grundlagen, Begrifflichkeiten, Sprachen und Methoden des Software Engineering vermittelt, die dann in den begleitenden Übungen (Zentralübung, Präsenz- und/oder Heimübungen in Kleingruppen) vertieft und in dem begleitenden Praktikumsanteil im Team an einem durchgängigen Beispiel von den Studierenden selbst erprobt werden. Hierbei kommen gängige Software-Engineering-Werkzeuge zum Einsatz.                  Lernmaterialien, Literaturangaben:</p> <ul style="list-style-type: none"> <li>Präsentationsfolien, ggf. Tafelanschrieb, i.d.R. Vorlesungsaufzeichnung</li> <li>Übungsaufgaben und Lösungshinweise</li> <li>Praktikumsaufgabe mit Beispielartefakten</li> <li>Konkrete Literatur zu den einzelnen Themengebieten wird in der Veranstaltung bekanntgegeben.</li> </ul>								

<b>Informatik und Gesellschaft</b>							
Computer Science and Society							
<b>Modulnummer:</b>	<b>Workload (h):</b>	<b>LP:</b>	<b>Studiensemester:</b>	<b>Turnus:</b>	<b>Dauer (in Sem.):</b>	<b>Sprache:</b>	<b>P/WP:</b>
B9	150	5	6.	SoSe	1	de	P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Informatik und Gesellschaft	V Ü	45 30	75	P	100 25	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> keine						
<b>4</b>	<p><b>Inhalte:</b></p> <p>Die Veranstaltung gibt einen Überblick über die gesellschaftlichen Aspekte der Informationstechnik und versetzt die Studierenden in die Lage, die gesellschaftliche und ethische Bedeutung des Fachs zu beurteilen und Konsequenzen für verantwortungsbewusstes Handeln zu ziehen. Die Studierenden verstehen die Zusammenhänge zwischen Informationstechnik und Gesellschaft, sind in der Lage, die Auswirkungen informationstechnischer Produkte und Dienstleistungen zu analysieren und potentielle Konfliktfelder zu identifizieren, und kennen ethische Verhaltensregeln, wie sie zum Beispiel von den Berufsorganisationen GI, IEEE und ACM herausgegeben werden.</p> <p>Die Veranstaltung umfasst folgende Inhalte:</p> <ul style="list-style-type: none"> <li>• Geschichte der Informatik: Geschichte des Schreibens, Rechnens und Kommunizierens, Wechselwirkungen zwischen der Entwicklung von Gesellschaft und Technologien, Charakteristika und Potenziale des Digitalen, Informationstechnologie zwischen Herrschaftsinstrument oder Empowerment</li> <li>• Einführung in Ethik: Ethische Theorien, Theorie der Gerechtigkeit, Ethische Leitlinien</li> <li>• Vernetzte Kommunikation: Wissensgesellschaft, Informationsfreiheit, Meinungsfreiheit und Zensur, Chatkontrolle, Netzneutralität, Störerhaftung</li> <li>• Geistiges Eigentum: Urheberrecht, Digital Rights Management, Markenrecht, Patentrecht, Software als Produkt, Free and Open Software, Creative Commons</li> <li>• Datenschutz: Entstehung des Datenschutzes, Datenschutz vs. Privatsphäre, Grundkonzepte des Datenschutzes, Grundsätze der DSGVO, Rechte der Betroffenen, Technisch-organisatorische Maßnahmen, Auftragsverarbeitung, Datenschutz-Management</li> <li>• Datenschutzgefährdende Technologien: Datenspuren im Netz und in der realen Welt, Datenspeicherung im Ausland, Gesichtserkennung, Social Profiling, Predictive Policing, RFID-Technologie, Surveillance Capitalism, Big Data, De-Pseudonymisierung</li> <li>• Informatik und das Militär: Militärgeschichte der Informatik, Kernprobleme im militärischen Bereich, Besonderheiten von Software, Fehler und Zuverlässigkeit von Computern, Autonome Waffensysteme, Cyberwar</li> </ul>						
<b>5</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen:</b></p> <p>Die Studierenden können</p> <ul style="list-style-type: none"> <li>• ethische Herausforderungen in der Arbeit von Informatikerinnen und Informatikern erkennen,</li> <li>• Grundkonzepte der behandelten Themenfelder (Geschichte, Ethik, Datenschutz, geistiges Eigentum etc.) erklären,</li> <li>• Systemanforderungen auf Grundlage der ethischen und rechtlichen Rahmenbedingungen formulieren,</li> <li>• Wechselwirkungen zwischen rechtlichen Rahmenbedingungen und Informatiksystemen analysieren,</li> </ul>						

	<ul style="list-style-type: none"> <li>informatische Aspekte in gesellschaftlich relevanten Ereignissen bewerten.</li> </ul>		
6	<b>Prüfungsleistung:</b> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
	zu	Prüfungsform	Dauer bzw. Umfang
	a)	Klausur	90-120 Minuten
7	<b>Studienleistung / qualifizierte Teilnahme:</b>		
	zu	Form	Dauer bzw. Umfang
	a)	Übungsaufgaben	SL / QT
<p>Von der bzw. dem jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>			
8	<b>Voraussetzungen für die Teilnahme an Prüfungen:</b> Bestandene Studienleistung		
9	<b>Voraussetzungen für die Vergabe von Leistungspunkten:</b> Bestandene Modulabschlussprüfung		
10	<b>Gewichtung für Gesamtnote:</b> Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).		
11	<b>Verwendung des Moduls in anderen Studiengängen:</b> Dieses Modul findet auch Verwendung in den Studiengängen B.Sc. Informatik, B.Sc. Computer Engineering, B.Ed. HRSGe Informatik sowie im Studiengang B.Ed. BK Informatik.		
12	<b>Modulbeauftragte/r:</b> Dr. Harald Selke		
13	<b>Sonstige Hinweise:</b> <b>Methodische Umsetzung</b> Die Inhalte werden durch im Rahmen der Vorlesung präsentiert, wobei interaktive Elemente sowie punktuell das Flipped-Classroom-Konzept zur Anwendung kommen. In wöchentlich zu bearbeitenden Übungsaufgaben werden Fallbeispiele und Szenarien erarbeitet. In den Übungen diskutieren die Studierenden in Kleingruppen und referieren in den Übungen sowie in den Übungsaufgaben erarbeitete Positionen und Lösungsvorschläge.		
	Lernmaterialien, Literaturangaben: <ul style="list-style-type: none"> <li>Michael J. Quinn: Ethics for the Information Age. 8th edition, Pearson, 2019.</li> <li>Sara Baase, Timothy M. Henry: A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology. 5th edition, Pearson, 2018.</li> <li>Felix Winkelkemper: Interface Evolution – Die Geschichte des Computers als Geschichte seiner Nutzungsschnittstelle. Eigenverlag, 2021.</li> <li>Weitere Literatur wird in der Veranstaltung bekanntgegeben.</li> </ul>		

<b>Softwareprojekt (Lehramtsstudierende GyGe)</b>							
Software Project GyGe							
<b>Modulnummer:</b>	<b>Workload (h):</b>	<b>LP:</b>	<b>Studiensemester:</b>	<b>Turnus:</b>	<b>Dauer (in Sem.):</b>	<b>Sprache:</b>	<b>P/WP:</b>
B10	210	7	6.	SoSe	1	de	P
<b>1</b>	<b>Modulstruktur:</b>						
	<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>	
	Softwareprojekt für Lehramtsstudierende – GyGe	P	60	150	P	15	
<b>2</b>	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> keine						
<b>3</b>	<b>Teilnahmevoraussetzungen:</b> Empfohlen wird die Teilnahme an den Modulen Programmierung I und II, Datenstrukturen, Algorithmen und formale Sprachen sowie Software Engineering.						
<b>4</b>	<b>Inhalte:</b> Die Veranstaltung gibt einen Einblick in eine praxisorientierte Softwareentwicklungsaufgabe, bei der die Studierenden im Team von ca. 15 Studierenden unter Verwendung von UML und Java ein Softwareprojekt umsetzen, dokumentieren und reflektieren. Die Veranstaltung umfasst unter anderem folgende Inhalte: <ul style="list-style-type: none"> <li>• Realisierung eines schulbezogenen Softwareprojekts für eine Zielgruppe der Sek II</li> <li>• Nutzung von professionellen Softwarewerkzeugen zur Softwareentwicklung im Team</li> <li>• Einführung in Methoden des Projektmanagements und der Qualitätssicherung bei der Organisation von Softwareprojekten</li> <li>• Praktische Anwendung von Methoden der Softwareentwicklung</li> <li>• Vorgehensmodelle für den Entwurf großer Softwaresysteme</li> <li>• Methoden und Sprachen für den objektorientierten Entwurf</li> </ul>						
<b>5</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen:</b> <b>Fachliche Kompetenzen:</b> Die Studierenden <ul style="list-style-type: none"> <li>• können Techniken und Werkzeuge zur (objektorientierten) Modellierung, Dokumentation und Organisation größerer Softwareprojekte anwenden;</li> <li>• kennen fachwissenschaftliche und fachdidaktische Anforderungen an Softwareprojekte im Informatikunterricht an Gymnasien und Gesamtschulen können sie zur Unterrichtsgestaltung nutzen;</li> <li>• können Sprachen und Werkzeuge im Softwareentwicklungsprozess einsetzen sowie den organisatorischen Ablauf eines Softwareprojekts von der Anforderungsdefinition bis zur Abgabe praktisch gestalten;</li> <li>• verfügen über Planungskompetenz für die Organisation schulischer Softwareprojekte an Gymnasien und Gesamtschulen;</li> <li>• kennen die Probleme teamorientierter Softwareentwicklung und können Methoden zu ihrer Lösung anwenden;</li> <li>• können Softwareprojekte und Entwicklungstools im Hinblick auf ihre Tauglichkeit für den Einsatz im Informatikunterricht an Gymnasien und Gesamtschulen fachwissenschaftlich und fachdidaktisch beurteilen;</li> <li>• können die Kriterien der BITV (Verordnung zur Schaffung barrierefreier Informationstechnik) bei der Entwicklung und Analyse von Software anwenden.</li> </ul>						

	<b>Schlüsselkompetenzen</b> <ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>								
6	<b>Prüfungsleistung:</b> <input type="checkbox"/> Modulabschlussprüfung (MAP) <input checked="" type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 45%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 25%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td></td> <td>Projektarbeit</td> <td>ca. 30 Minuten</td> <td>100 %</td> </tr> </tbody> </table>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote		Projektarbeit	ca. 30 Minuten	100 %
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
	Projektarbeit	ca. 30 Minuten	100 %						
7	<b>Studienleistung / qualifizierte Teilnahme:</b> keine								
8	<b>Voraussetzungen für die Teilnahme an Prüfungen:</b> keine								
9	<b>Voraussetzungen für die Vergabe von Leistungspunkten:</b> Bestandene Modulprüfung								
10	<b>Gewichtung für Gesamtnote:</b> Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).								
11	<b>Verwendung des Moduls in anderen Studiengängen:</b> Keine								
12	<b>Modulbeauftragte/r:</b> Prof. Dr. Carsten Schulte								
13	<b>Sonstige Hinweise:</b> Dieses Modul beinhaltet die Auseinandersetzung mit inklusionsrelevanten Fragestellungen im Umfang eines Workloads von 1 LP.								



Schlüsselqualifikation							
Key Qualification							
Modulnummer:	Workload (h):	LP:	Studiensemester:	Turnus:	Dauer (in Sem.):	Sprache:	P/WP:
S1	120	4	5.	Jedes Semester	1	de	P
1	<b>Modulstruktur:</b>						
		<b>Lehrveranstaltung</b>	<b>Lehrform</b>	<b>Kontaktzeit (h)</b>	<b>Selbststudium (h)</b>	<b>Status (P/WP)</b>	<b>Gruppengröße (TN)</b>
		Proseminar Informatik	PS	30	90	WP	15
2	<b>Wahlmöglichkeiten innerhalb des Moduls:</b> Für das Proseminar können alle Proseminare aus dem Angebot des Bachelorstudiengangs Informatik gewählt werden.						
3	<b>Teilnahmevoraussetzungen:</b> Empfohlen wird die Teilnahme an den Modulen, die zum gewählten Seminarthema passen.						
4	<b>Inhalte:</b> Im Proseminar soll beispielhaft die Einarbeitung in ein wissenschaftliches Thema erlernt und abstraktes Denken gestärkt werden. Die Inhalte sollen schriftlich und mündlich präsentiert werden. Dazu soll Basiswissen in Bezug auf Literaturrecherche, Rhetorik und aktuelle Präsentationstechniken sowie in Bezug auf Kritikfähigkeit und Feedbackmethoden erworben und angewendet werden.						
5	<b>Lernergebnisse (learning outcomes) / Kompetenzen:</b> <b>Fachliche Kompetenzen:</b> Im Proseminar werden neben dem inhaltlichen Aspekt vor allem das Aufbereiten eines Themas und seine Präsentation eingeübt. Die Studierenden lernen in der praktischen Durchführung das Erarbeiten eines Themas, das Treffen von Auswahlen, das Halten von Vorträgen, den Umgang mit Fragen und Diskussionsbeiträgen sowie das Anfertigen von größeren schriftlichen Texten. Die erarbeiteten Kompetenzen im Proseminar bereiten das Bewältigen ähnlicher Situationen später im Studium (Seminar, Projektgruppe, Abschlussarbeit) und im Beruf (Präsentation, Berichte) vor. <b>Schlüsselkompetenzen:</b> <ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Einsatz und Engagement</li> <li>• Haltung und Einstellung</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Medienkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>						
6	<b>Prüfungsleistung:</b> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)						
	<b>zu</b>	<b>Prüfungsform</b>	<b>Dauer bzw. Umfang</b>			<b>Gewichtung für die Modulnote</b>	
		Referat mit Ausarbeitung	45–60 Minuten			100 %	

7	<b>Studienleistung / qualifizierte Teilnahme:</b> keine
8	<b>Voraussetzungen für die Teilnahme an Prüfungen:</b> keine
9	<b>Voraussetzungen für die Vergabe von Leistungspunkten:</b> Bestandene Modulabschlussprüfung
10	<b>Gewichtung für Gesamtnote:</b> Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).
11	<b>Verwendung des Moduls in anderen Studiengängen:</b> Das Modul findet Verwendung im Studiengang B. Ed. BK Informatik.
12	<b>Modulbeauftragte/r:</b> Prof. Dr. Carsten Schulte
13	<b>Sonstige Hinweise:</b> keine

## Artikel II

- (1) Diese Änderungssatzung tritt am 1. Oktober 2025 in Kraft.
- (2) Diese Besonderen Bestimmungen gelten für alle Studierenden, die ab dem Wintersemester 2025/26 erstmalig für den Bachelorstudiengang Lehramt an Gymnasien und Gesamtschulen mit dem Unterrichtsfach Informatik an der Universität Paderborn eingeschrieben werden. Studierende, die bereits vor dem Wintersemester 2025/26 an der Universität Paderborn für den Bachelorstudiengang Lehramt an Gymnasien und Gesamtschulen mit dem Unterrichtsfach Informatik eingeschrieben worden sind, legen ihre Bachelorprüfung einschließlich Wiederholungsprüfungen letztmalig im Wintersemester 2029/30 nach den Besonderen Bestimmungen in der Fassung vom 31. Mai 2022 (AM.Uni.Pb 75.22) ab. Ab dem Sommersemester 2030 wird die Bachelorprüfung einschließlich Wiederholungsprüfungen nach diesen Besonderen Bestimmungen abgelegt.
- (3) Studierende können auf Antrag in diese Besonderen Bestimmungen wechseln. Studierende können nicht zurückwechseln.
- (3) Diese Änderungssatzung wird in den Amtlichen Mitteilungen der Universität Paderborn (AM.Uni.Pb) veröffentlicht.
- (4) Gemäß § 12 Absatz 5 HG kann nach Ablauf eines Jahres seit der Bekanntmachung dieser Ordnung gegen diese Ordnung die Verletzung von Verfahrens- oder Formvorschriften des Hochschulgesetzes oder des Ordnungs- oder des sonstigen autonomen Rechts der Hochschule nicht mehr geltend gemacht werden, es sei denn,
  1. die Ordnung ist nicht ordnungsgemäß bekannt gemacht worden,
  2. das Präsidium hat den Beschluss des die Ordnung beschließenden Gremiums vorher beanstandet,
  3. der Form- oder Verfahrensmangel ist gegenüber der Hochschule vorher gerügt und dabei die verletzte Rechtsvorschrift und die Tatsache bezeichnet worden, die den Mangel ergibt, oder
  4. bei der öffentlichen Bekanntmachung der Ordnung ist auf die Rechtsfolge des Rügeausschlusses nicht hingewiesen worden.

Ausgefertigt aufgrund des Beschlusses des Fakultätsrats der Fakultät für Elektrotechnik, Informatik und Mathematik vom 12. Mai 2025 im Benehmen mit dem Zentrumsrat der PLAZ – Professional School of Education vom 17. April 2025 sowie nach Prüfung der Rechtmäßigkeit durch das Präsidium der Universität Paderborn vom 14. Mai 2025.

Paderborn, den 4. Juli 2025

Der Präsident  
der Universität Paderborn

Professor Dr. Matthias Bauer

---

**HERAUSGEBER  
PRÄSIDIUM DER UNIVERSITÄT PADERBORN  
WARBURGER STR. 100  
33098 PADERBORN**

**[HTTP://WWW.UNI-PADERBORN.DE](http://www.uni-paderborn.de)**

---

**ISSN 2199-2819**