

<b>Modulcode</b> (1.)	<b>Modulbezeichnung</b> (2.)	<b>Zuordnung</b> (3.)
BAI1040	Grundkonzepte der Programmierung (GKP)	
	<b>Studiengang</b> (4.)	Bachelor Angewandte Informatik/ Bachelor Angewandte Informatik DUAL
	<b>Fakultät</b> (5.)	Gebäudetechnik und Informatik

<b>Modulverantwortlich</b> (6.)	Prof. Dr.-Ing. Jörg Sahm
<b>Modulart</b> (7.)	Pflicht
<b>Angebotshäufigkeit</b> (8.)	WS
<b>Regelbelegung / Empf. Semester</b> (9.)	BA1
<b>Credits (ECTS)</b> (10.)	5 CP
<b>Leistungsnachweis</b> (11.)	PL (N)
<b>Unterrichtssprache</b> (12.)	Deutsch
<b>Voraussetzungen für dieses Modul</b> (13.)	-
<b>Modul ist Voraussetzung für</b> (14.)	BAI2040: Objektorientierte Programmierung BAI3010: Programmierung Java 1 BAI4010: Programmierung Java 2 BAI6010: Programmierung mobiler Endgeräte BAI6120: Grafische Datenverarbeitung 1 BAI7130: Grafische Datenverarbeitung 2 BAI6530: Geo-Informationssysteme BAI5530: Einführung Künstliche Intelligenz
<b>Moduldauer</b> (15.)	1 Semester
<b>Notwendige Anmeldung</b> (16.)	-
<b>Verwendbarkeit des Moduls</b> (17.)	Sämtliche Fächer, in denen grundlegende Programmierkompetenzen benötigt werden

	<b>Lehrveranstaltung</b> (18.)	<b>Dozent/in</b> (19.)	<b>Art</b> (20.)	<b>Teilnehmer (maximal)</b> (21.)	<b>Anzahl Gruppen</b> (22.)	<b>SWS</b> (23.)	<b>Workload</b>	
							<b>Präsenz</b> (24.)	<b>Selbststudium</b> (25.)
1	Grundkonzepte der Programmierung	Sahm	V	100	1	2	30	15
2	Grundkonzepte der Programmierung	Sahm	Ü	25	4	2	30	50
<b>Summe</b>						<b>4</b>	<b>60</b>	<b>65</b>
<b>Workload für das Modul</b> (26.)							<b>125</b>	

<b>Qualifikationsziele</b> <span style="float: right;">(27.)</span>	<p>Die Studierenden ...</p> <ul style="list-style-type: none"> <li>• zerlegen komplexere verbale Aufgabenstellungen in Teilaufgaben und ordnen diese ggf. in Hierarchien</li> <li>• stellen Abläufe in Flussdiagrammen dar</li> <li>• verstehen die grundlegende Rechnerarchitektur</li> <li>• kennen die Aufgaben und Ausgaben eines Compilers</li> <li>• implementieren Algorithmen und Programme in einer integrierten Entwicklungsumgebung</li> <li>• repräsentieren Informationen in Form primitiver und komplexer Datentypen</li> <li>• verstehen die Kommunikation von Programmteilen und die Bedeutung von Adressen bzw. Pointern für diese Kommunikation</li> <li>• organisieren Daten in Containern wie Array, Listen, Bäumen</li> <li>• sortieren und suchen Daten in Containern</li> <li>• verstehen boolesche Ausdrücke und verwenden diese in Kontrollstrukturen wie etwa Bedingungen und Schleifen</li> <li>• gliedern Programme in Methoden und Dateien</li> </ul>
<b>Inhalte</b> <span style="float: right;">(28.)</span>	<ul style="list-style-type: none"> <li>• Teile-und-herrsche-Verfahren</li> <li>• Flussdiagramme</li> <li>• Von-Neumann-Architektur</li> <li>• Befehlsabarbeitung</li> <li>• Compilerablauf</li> <li>• Integrierte Entwicklungsumgebungen (Visual Studio)</li> <li>• Primitive Datentypen</li> <li>• Komplexe Datentypen</li> <li>• Pointer</li> <li>• String</li> <li>• Gültigkeitsbereiche und Lebensdauer von Variablen</li> <li>• Dynamisches Reservieren und Freigeben von Speicher</li> <li>• Funktionen und Prozeduren</li> <li>• Array und Stack</li> <li>• Einfach und doppelt verkettete Listen</li> <li>• Bäume als rekursive Datenstrukturen</li> <li>• Rekursive Abarbeitung von Bäumen mittels DSF</li> <li>• Effektive Sortierung und Suche in Datenstrukturen</li> <li>• Kontrollstrukturen</li> <li>• Softwaregliederung in mehrere Dateien</li> </ul>
<b>Vorleistungen und Modulprüfung</b> <span style="float: right;">(29.)</span>	<p>Vorleistungen:</p> <ul style="list-style-type: none"> <li>• keine</li> </ul> <p>Modulprüfung:</p> <ul style="list-style-type: none"> <li>• 100% Klausur (120 min) im Prüfungszeitraum</li> </ul>
<b>Literatur</b> <span style="float: right;">(30.)</span>	<ul style="list-style-type: none"> <li>• S. B. Lippman, J. Lajoie, B. E. Moo: C++ Primer</li> <li>• B. Stroustrup: The C++ Programming Language</li> <li>• B. Stroustrup: Programming: Principles and Practice Using C++</li> <li>• S. Meyers: Effective C++</li> <li>• M. T. Goodrich, R. Tamassia, D. M. Mount Data Structures and Algorithms in C++</li> <li>• R. Sedgewick: Algorithms</li> </ul>