

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

Modulverantwortlich (6.)	Prof. Dr.-Ing. Jörg Sahm
Modulart (7.)	Pflicht
Angebotshäufigkeit (8.)	WS
Regelbelegung / Empf. Semester (9.)	BA1
Credits (ECTS) (10.)	5 CP
Leistungsnachweis (11.)	PL (N)
Unterrichtssprache (12.)	Deutsch
Voraussetzungen für dieses Modul (13.)	-
Modul ist Voraussetzung für (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
Moduldauer (15.)	1 Semester
Notwendige Anmeldung (16.)	-
Verwendbarkeit des Moduls (17.)	Sämtliche Fächer, in denen grundlegende Programmierkompetenzen benötigt werden

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

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