

**Kommunikations-, Informations- und Mikrotechnik (KIM), Energiesysteme und Automation (ESA), Informatik / Softwaretechnik**

Modulbezeichnung	<b>Programmieren II</b>
Kürzel für Stundenplan	ProgII
Semester	2
Modulverantwortliche(r)	Prof. Dipl.-Ing. Christian Ebel
Dozent(in)	Prof. Dipl.-Ing. Christian Ebel
Sprache	Deutsch
Zuordnung zum Curriculum	ESA, KIM und INF
Lehrform / SWS	<p>4 V, Gruppengröße ca. 100                  2 P, Gruppengröße max. 12</p> <p>Vorlesung mit begleitendem Praktikum. Im Rahmen des Praktikums werden insgesamt fünf auf den Vorlesungsfortschritt angepasste Aufgabenstellungen bearbeitet. Die detaillierten Aufgabenstellungen werden von den Studierenden eigenständig bearbeitet. Jede Aufgabe wird mit der/dem Studierenden individuell besprochen und bewertet. Es werden Teams mit jeweils zwei Studierenden gebildet.</p>
Arbeitsaufwand	<p>64 h Vorlesung                  32 h Vor-/Nachbereitung Vorlesung                  84 h Praktikum</p>
Kreditpunkte	<p>6                  Die Notenfindung erfolgt durch die Bewertung der fünf Praktikumsaufgaben</p>
Voraussetzungen	<p>Modul Programmieren I                  (Kenntnisse der Programmiersprache C)</p>
Lernziele / Kompetenzen	<p>Die Studierenden lernen und üben die Tätigkeiten zum objektorientierten Lösen einer Aufgabenstellung durch Software.</p> <p>Am Beispiel der Programmiersprache C++ werden unterschiedliche Aufgabenstellungen bearbeitet. Bei der Umsetzung einer Aufgabe wird konsequent auf einen methodisch strukturierten "Top→Down" Ansatz geachtet.</p> <ul style="list-style-type: none"> <li>- Umsetzung der Aufgabenstellung in ein Objektmodell.</li> <li>- Modellierung von Klassen</li> <li>- Umsetzung in die Sprache C++</li> <li>- Kodierung der Aufgabenstellung</li> <li>- Programmtest unter Beachtung von Randbedingungen</li> </ul> <p>Die Studierenden kennen die Programmiersprache C++. Sie sind in der Lage eine Aufgabenstellung zu analysieren, die gewählte Lösung zu dokumentieren und in objektorientierte Software umzusetzen.</p>

Inhalt	siehe Folgeseite
Studien- Prüfungsleistungen	Projektarbeit
Medienformen	Tafel, Overheadfolien, Beamer, Script
Literatur	<p>Christian Ebel, Vorlesungs-Script "Programmieren II", Hausdruckerei der Fachhochschule Lübeck</p> <p>Nicolai Josuttis, "Objektorientiertes Programmieren in C++", ISBN 3-8273-1771-1, Addison Wesley</p> <p>Peter Prinz, Ulla Kirch-Prinz, "C++ Lernen und professionell anwenden", ISBN 3-8266-0423-7, MITP-Verlag</p> <p>Bjarne Stroustrup, "Die C++ Programmiersprache", ISBN 0-201-53992-6, Addison Wesley</p> <p>Herbert Schildt, "C++ Ent-Packt", ISBN 3-8266-0731-7, MITP-Verlag</p>

Studieninhalte des Moduls <b>Programmieren II</b>	
<p>C++ eine Erweiterung der Sprache C</p> <ul style="list-style-type: none"> <li>- ergänzende Sprachdefinitionen</li> </ul>	
<p>Einführung in die Begriffe der objektorientierten Programmierung</p> <ul style="list-style-type: none"> <li>- Klasse, Objekt, Attribut, Methode, Vererbung, Polymorphismus</li> </ul>	
<p>Die Sprache C++</p> <ul style="list-style-type: none"> <li>- Übertragung objektorientierter Begriffe in die Programmiersprache C++</li> </ul>	
<p>Begriffsbestimmung und Anwendung an Beispielen</p> <ul style="list-style-type: none"> <li>- Datenkapselung, Konstruktoren, Destruktor, Operatorfunktionen, Überladung von Funktionen und Operatoren, Einfach- und Mehrfachvererbung, virtuelle Methoden, abstrakte Klassen, Polymorphie, Templates</li> <li>- Optimierung von Klassenentwürfen unter den Gesichtspunkten Codekonsistenz und Laufzeit</li> </ul>	
<p>Exception-Handling</p> <ul style="list-style-type: none"> <li>- Klassische Prinzipien der Fehlererkennung und Fehlerbehandlung</li> <li>- Das objektorientierte Konzept der Fehlerbehandlung, Entwurf von Fehlerklassen</li> </ul>	
<p>Bibliotheken</p> <ul style="list-style-type: none"> <li>- Einsatz von standardisierten Bibliotheken zur effektiven Erstellung von Software</li> <li>- Erstellung eigener Software-Bibliotheken</li> </ul>	
Summe Workload für die Vorlesung	64 h
Summe Workload Vor-/Nachbereitung Vorlesung	32 h
Summe Workload für das Praktikum	84 h
Gesamt-Workload für das Modul "Programmieren II"	180 h