

<b>Programmiersprachen</b> <i>Programming Languages</i>								Modulnummer:	
Bachelor Pflicht/Wahl <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/> Wahl <input type="checkbox"/> Sonderfall <input type="checkbox"/>				Modulbereich: Pflicht					
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	$\Sigma$	Kreditpunkte: 6	Turnus i. d. R. angeboten alle 2 Jahre
	0	0	4	0	0	0	4		
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Praktische Informatik 1, Praktische Informatik 2, Praktische Informatik 3									
Vorgesehenes Semester: ab 1. Semester									
Sprache: Deutsch									
Ziele: <ul style="list-style-type: none"> <li>• Beschreibungen von Programmiersprachen verstehen und in Hinblick auf Konzepte, auf die Unterstützung von Programmier-Paradigmen und auf Entwurfsziele analysieren können</li> <li>• Ausprägungen von Konzepten und Paradigmen in verschiedenen Programmiersprache vergleichen und bewerten können</li> <li>• hinterfragen, wie weit Programmiersprachen ein Programmierparadigma unterstützen und die von ihren Entwerfern gesteckten Entwurfsziele erreichen</li> </ul>									
Inhalte: Konzepte <ul style="list-style-type: none"> <li>• Werte (Datenstrukturen und Ausdrücke).</li> <li>• Speicher (Variablen und Befehle)</li> <li>• Bindung (Vereinbarungen und Gültigkeitsbereiche).</li> <li>• Abstraktion (Funktionen, Prozeduren und Parameterübergabe).</li> <li>• Kapselung (Moduln, abstrakte Datentypen, Klassen, generische Pakete).</li> <li>• Typsysteme (Überladen, Anpassungen, Polymorphie, Untertypen und Vererbung).</li> <li>• Ablaufsteuerung (Sprünge, Ausweg, Ausnahmen).</li> <li>• Nebenläufigkeit und Verteiltheit</li> </ul> Paradigmen (Programmierstile) <ul style="list-style-type: none"> <li>• Imperatives Programmieren.</li> <li>• Objekt-orientiertes Programmieren.</li> <li>• Nebenläufiges Programmieren.</li> <li>• Funktionales Programmieren.</li> <li>• Logisches Programmieren.</li> </ul> Prinzipien des Sprachentwurfs <ul style="list-style-type: none"> <li>• Syntax.</li> <li>• Semantik.</li> <li>• Pragmatik.</li> </ul> In der Übung Anwendung der in der Vorlesung erworbenen Kenntnisse und Fähigkeiten bei der Untersuchung spezifischer Konzepte und Eigenschaften von spezifischer Programmiersprachen (z. B. Ada, Eiffel, Java, Haskell, Prolog)									

Unterlagen (Skripte, Literatur, Programme usw.):

- David A. Watt: Programming Language Design Concepts, Chichester: Wiley and Sons (2004).
- Robert W. Sebesta: Concepts of Programming Languages 5/e, Reading, MA: Addison-Wesley (2002).

Weiteres Lehrmaterial ist auf der Webseite der Veranstaltung zu finden:

- Online-Fassung des Buches David A. Watt: Programmiersprachen - Konzepte und Paradigmen, München-Wien: Hanser (1996)
- Folienkopien
- Übungsaufgaben
- Beschreibungen der Referenzsprachen Ada, Eiffel, Java, Haskell, Prolog
- Hinweise auf Quellen im WWW

Implementierungen der Referenzsprachen Ada, Eiffel, Java, Haskell, Prolog stehen im Rechnernetz des Studiengangs zur Verfügung.

Form der Prüfung:

i.d.R. mündliche Prüfung

Arbeitsaufwand	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h

Lehrende:

Dr. B. Hoffmann

Verantwortlich:

Dr. B. Hoffmann