03-IMAP-PS	Programmiersprachen
	Programming Languages

Lehrform *(teaching format)* / **SWS** *(hours per week)*: 4K

Kreditpunkte *(credit points)*: 6

Turnus *(frequency)*: i.d.R. jedes SoSe

Inhaltliche Voraussetzungen *(content-related prior knowledge/skills)*: Praktische Informatik 1+2+3

Sprache *(language)*: Deutsch

Lehrende *(teaching staff)*: Prof. Dr. Christoph Lüth

Studiengang *(degree program)*	Module	Semester
Infornatik (Master)	IMAP, IMVP-SQ	ab 1.Sem.
Infornatik (Bachelor)	(nur *Freie Wahl*)	

Lernergebnisse:

- Beschreibungen von Programmiersprachen verstehen und in Hinblick auf Konzepte, auf die Unterstützung von Programmier-Paradigmen und auf Entwurfsziele analysieren können
- Ausprägungen von Konzepten und Paradigmen in verschiedenen Programmiersprache vergleichen und bewerten können
- Hinterfragen, wie weit Programmiersprachen ein Programmierparadigma unterstützen und die von ihren Entwerfern gesteckten Entwurfsziele erreichen

Learning Outcome:

Inhalte:

Konzepte

- Werte (Datenstrukturen und Ausdrücke).
- Speicher (Variablen und Befehle)
- Bindung (Vereinbarungen und Gültigkeitsbereiche).
- Abstraktion (Funktionen, Prozeduren und Parameterübergabe).
- Kapselung (Moduln, abstrakte Datentypen, Klassen, generische Pakete).
- Typsysteme (Überladen, Anpassungen, Polymorphie, Untertypen und Vererbung).
- Ablaufsteuerung (Sprünge, Ausweg, Ausnahmen).
- Nebenläufigkeit und Verteiltheit

Paradigmen (Programierstile)

- Imperatives Programmieren.
- Objekt-orientiertes Programmieren.
- Nebenläufiges Programmieren.
- Funktionales Programmieren.
- Logisches Programmieren.

Prinzipien des Sprachentwurfs

- Syntax.
- Semantik.
- Pragmatik.

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)

Contents:

Hinweise *(remarks)*: In der Tabelle sind nur die primären/spezifischsten Module aufgelistet, denen diese Veranstaltung zugeordnet ist.