Modulbezeichnung	**Praktische Informatik 3**
Modulverantwortliche(r)	Dr. B. Hoffmann
Modulart	Pflicht/Wahl ☒ Wahlpflicht ☐
Spezialisierungsbereich	
Dauer des Moduls	1 Semester
Kreditpunkte	6 CP
Arbeitsaufwand	Berechnung des Workloads Präsenz 56 h Übungsbetrieb/Prüfungsvorbereitung 124 h Summe 180 h
Turnus des Moduls	angeboten in jedem WiSe
Voraussetzung für die Teilnahme	Keine ☐ Folgende Inhaltliche Voraussetzungen: Praktische Informatik 2
Lehr- und Lernformen	Seminar ☐ Vorlesung ☒ Tutorium ☒ Praktikum ☐ Projekt ☐
Lernziele	• Konzepte und typische Merkmale des funktionalen Programmierens kennen, verstehen und anwenden können. • Datenstrukturen und Algorithmen in einer funktionalen Programmiersprache umsetzen und auf einfachere praktische Probleme anwenden können. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. Die Vorlesung Praktische Informatik 3 vermittelt essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik Voraussetzung ist.
Lerninhalte	1. Grundlagen der funktionalen Programmierung: Rekursion – Definition von Funktionen durch rekursive Gleichungen und Mustervergleich (pattern matching) – Auswertung, Reduktion, Normalform – Funktionen höherer Ordnung, currying,Typkorrektheit und Typinferenz 2. Typen: Algebraische Datentypen – Typkonstruktoren – Typklassen – Polymorphie – Standarddatentypen (Listen, kartesische Produkte, Lifting) und Standardfunktionen darauf (fold, map, filter) – Listenkomprehension 3. Algorithmen und Datenstrukturen: Unendliche Listen (Ströme) – Bäume – Graphen – zyklische Datenstrukturen 4. Strukturierung und Spezifikation: Module – Schnittstellen (Interfaces) – Abstrakte Datentypen – Signaturen und Axiome 5. Theoretische Aspekte: Referentielle Transparenz – Lambda-Kalkül –- Beweis durch Induktion 6. Fortgeschrittene Funktionale Programmierung: Funktionale I/O und zustandsbasierte Prorgamme – Monaden Im Übungsbetrieb; Programmentwicklung in Haskell —- Realisierung einzelner, überschaubarer Programmieraufgaben in kleinen Gruppen
Prüfungsformen	i. d. R. Bearbeitung von Übungsaufgaben und Klausur

Literatur	• Simon Thompson: Haskell - The Craft of Functional Programming, Addison-Wesley, 3. Auflage 2011. • Peter Pepper: Funktionale Programmierung. Springer-Verlag 1999. Weiteres Lehrmaterial ist auf der Webseite des Veranstaltung zu finden: • Folienkopien • Übungsaufgaben • Hinweise auf Quellen im WWW Das Haskell-System ghci ist frei verfügbare Software (für Linux, Windows und MacOS).