

#### 4.1.5 Informatik 1

<b>Informatik 1</b>
<b>Modulübersicht</b>
EDV-Bezeichnung: QUCB150
Modulverantwortliche(r): Prof. Dr. NN
Modulumfang (ECTS): 6 CP
Einordnung (Semester): 1. Semester
Inhaltliche Voraussetzungen: inhaltlich keine, Kenntnisse in der Bedienung eines PCs werden vorausgesetzt
Voraussetzungen nach SPO: Nach SPO sind keine formellen Voraussetzungen erforderlich.
<p>Lernergebnisse und Kompetenzen: Die Studierenden erwerben grundlegende Kenntnisse der Informatik und der Programmierung. Sie können einfache algorithmische Probleme analysieren, modellieren und in einer höheren Programmiersprache umsetzen.</p> <p><b>Fachliche Kompetenzen</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• wenden grundlegende Programmierkonstrukte (Variablen, Kontrollstrukturen, Arrays) an,</li> <li>• entwerfen und implementieren einfache objektorientierte Strukturen (Klassen, Objekte, Methoden),</li> <li>• nutzen grundlegende Datenstrukturen und Rekursion zur Problemlösung,</li> <li>• führen einfache Laufzeitanalysen mit dem O-Kalkül durch,</li> <li>• formulieren und analysieren Probleme mit Aussagenlogik und elementaren formalen Methoden (z. B. endliche Automaten).</li> </ul> <p><b>Methodische Kompetenzen:</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• entwickeln und testen kleine Programme mit geeigneten Entwicklungswerkzeugen,</li> <li>• wenden strukturierte Vorgehensweisen zur Fehlersuche und Qualitätssicherung an,</li> <li>• beurteilen die Effizienz und Korrektheit ihrer Lösungen.</li> </ul> <p><b>Sozial- und Selbstkompetenzen:</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• arbeiten selbstständig und im Team an Programmieraufgaben,</li> <li>• planen und reflektieren ihren Lern- und Arbeitsprozess,</li> <li>• dokumentieren und präsentieren Ergebnisse adressatengerecht.</li> </ul>
<p>Prüfungsleistungen: Informatik Grundlagen: Klausur, 120 Minuten Übungen Informatik 1 (Studienleistung): Die Übungen gelten als bestanden, wenn die Übungsblätter/ -projekte erfolgreich bearbeitet und testiert wurden.</p>

<b>Lehrveranstaltung: Informatik 1</b>
EDV-Bezeichnung: QUCB151
Dozierende(r): Prof. Dr. NN
Umfang (SWS): 4 SWS
Turnus: Wintersemester
Art, Modus: Vorlesung, Pflichtfach
Lehrsprache: Deutsch
<p>Studieninhalte:</p> <ul style="list-style-type: none"> <li>• Einführung in die Informatik und Programmierparadigmen</li> <li>• Aufbau, Syntax und Semantik höherer Programmiersprachen (C++ / Python)</li> <li>• Grundlegende Programmierkonstrukte: Datentypen, Variablen, Operatoren, Kontrollstrukturen, Arrays</li> <li>• Funktionen, Parameterübergabe, Rekursion, Divide-and-Conquer</li> <li>• Objektorientierte Grundlagen: Klassen, Objekte, Methoden, Kapselung, einfache Vererbung</li> <li>• Einführung in grundlegende Datenstrukturen (z. B. Listen, Stacks, Queues)</li> <li>• Prinzipien der Softwarequalität: Modularisierung, Code-Konventionen, Dokumentation, Testen</li> <li>• Grundlagen der Laufzeitanalyse (O-Kalkül)</li> <li>• Formale Grundlagen: Aussagenlogik, Beweistechniken, formale Sprachen und endliche Automaten</li> </ul>
<p>Empfohlene Literatur/Entwicklungssoftware:</p> <ul style="list-style-type: none"> <li>• Bjarne Stroustrup: The C++ Programming Language, 4. Aufl., Addison-Wesley, 2013.</li> <li>• Bernd Klein: Einführung in Python 3: Für Ein- und Umsteiger, 4. Aufl., Hanser, 2021.</li> <li>• D. W. Hoffmann: Theoretische Informatik, 5. Aufl., Hanser, 2022.</li> <li>• Michael Sipser: Introduction to the Theory of Computation, 3. Aufl., Cengage Learning, 2012.</li> </ul>

<b>Lehrveranstaltung: Übungen Informatik 1</b>
EDV-Bezeichnung: QUCB152
Dozierende(r): Prof. Dr. NN
Umfang (SWS): 2 SWS
Turnus: Wintersemester
Art, Modus: Übungen, Pflichtfach
Lehrsprache: Deutsch
<p>Studieninhalte:</p> <ul style="list-style-type: none"> <li>• Implementieren und Testen kleiner Programme zur Vertiefung der Vorlesungsinhalte</li> <li>• Praktische Übungen zu Variablen, Kontrollstrukturen und Arrays</li> <li>• Entwicklung objektorientierter Programme mit Klassen, Objekten und Methoden</li> </ul>

- Einsatz von Rekursion und Divide-and-Conquer-Verfahren in Programmieraufgaben
- Anwendung einfacher Datenstrukturen (Listen, Stacks, Queues)
- Debugging und Fehlersuche mit einer integrierten Entwicklungsumgebung (IDE)
- Nutzung von Versionsverwaltung und Testwerkzeugen
- Kleine praxisnahe Programmierprojekte in Kleingruppen

Empfohlene Literatur/Entwicklungssoftware:

- Siehe Vorlesung Informatik 1.
- PhyCharm, CLion