

<b>03-IBAT-KS</b>	<b>Korrekte Software: Grundlagen und Methoden</b>
	<i>Correct Software: Foundations and Methods</i>

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

**Kreditpunkte** (*credit points*): 6

**Turnus** (*frequency*): i.d.R. alle 2 Jahre

**Inhaltliche Voraussetzungen** (*content-related prior knowledge/skills*): Elementare Programmierkenntnisse

**Sprache** (*language*): Deutsch / English

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

<b>Studiengang</b> ( <i>degree program</i> )	<b>Module</b>	<b>Semester</b>
Informatik (Bachelor VF)	IBAT	ab 4. Sem.
Informatik (Bachelor KF)	KINF-A1/A2	ab 4. Sem.
Systems Engineering (Bachelor)	V07-ESS-V	ab 4. Sem.
Informatik (Master)	<i>General Studies</i>	ab 1. Sem.
Systems Engineering II (Master)	M07-AM-INF	1. Sem.

### **Lernergebnisse:**

Ziel der Veranstaltung ist es, die Grundbegriffe der korrekten Softwareentwicklung zu verstehen. Wie können wir Software schreiben, die tut was sie soll? Und wie können wir das beweisen?

Dazu betrachten wir die Grundbegriffe der formalen Logik und Semantik, und insbesondere der Floyd-Hoare-Logik. Lernziel ist es, Eigenschaften von einfachen C-Programmen spezifizieren und beweisen zu können, und zu verstehen, wie diese Techniken auf reale C-Programme (oder andere Programmiersprachen) skalieren können.

*Learning Outcome:*

A successful participant of this module will learn the fundamentals of correct software development. How can we write software which does what it is supposed to do? How can we specify that, and how can we prove it?

### **Inhalte:**

- Logische Grundlagen: Formale Logik, Prädikatenkalkül, Vollständigkeit und Korrektheit;
- Grundlagen der Floyd-Hoare-Logik;
- Operationale und denotationale Semantik für eine einfach imperative Programmiersprache;
- Vollständigkeit und Korrektheit der Floyd-Hoare-Logik für diese Sprache;
- Berechnung von Verifikationsbedingungen;
- Erweiterung der Programmlogik um strukturierte Datentypen, Funktionsaufrufe, und Referenzen (Zeiger);

*Contents:*

- Foundations: formal logic, propositional and predicate logic, soundness and completeness;
  - Foundations of the Floyd-Hoare-Logic;
  - Operational and denotational semantics for a simple imperative language;
  - Soundness and completeness for Floyd-Hoare-logic for this language;
  - Generation of verification conditions;
  - Extending the program logic with structured datatypes, function calls, and references (pointers).
- 

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