HOCHSCHULE ESSLINGEN

Nah an Mensch und Technik.

Fakultät Informatik und Informationstechnik

Modulhandbuch SPO1

Bachelor-Studiengang
IT-Sicherheit (ISB)

Hinweise:

Die in den Modulbeschreibungen genannten Voraussetzungen sind nicht zwingend, aber sehr hilfreich für das Verständnis der vermittelten Lerninhalte.

Alle Module haben eine Dauer von einem Semester.

Die Verwendung der Module in den verschiedenen Studiengängen ist an der Nennung der Studiengänge in der Zielgruppe zu erkennen,

Abkürzungen:

SWS Semesterwochenstunden
ECTS European Credit Transfer and Accumulation System
Europäisches System zur Übertragung und Akkumulierung von Studienleistungen

ECTS ist ein Maß für den erforderlichen Arbeitsaufwand im Studium (Workload)
1 ECTS entspricht näherungsweise 30 Arbeitsstunden

Die Angabe der ECTS-Punkte in den Modulbeschreibungen soll den aufzubringenden Workload transparent machen.

Studiengänge

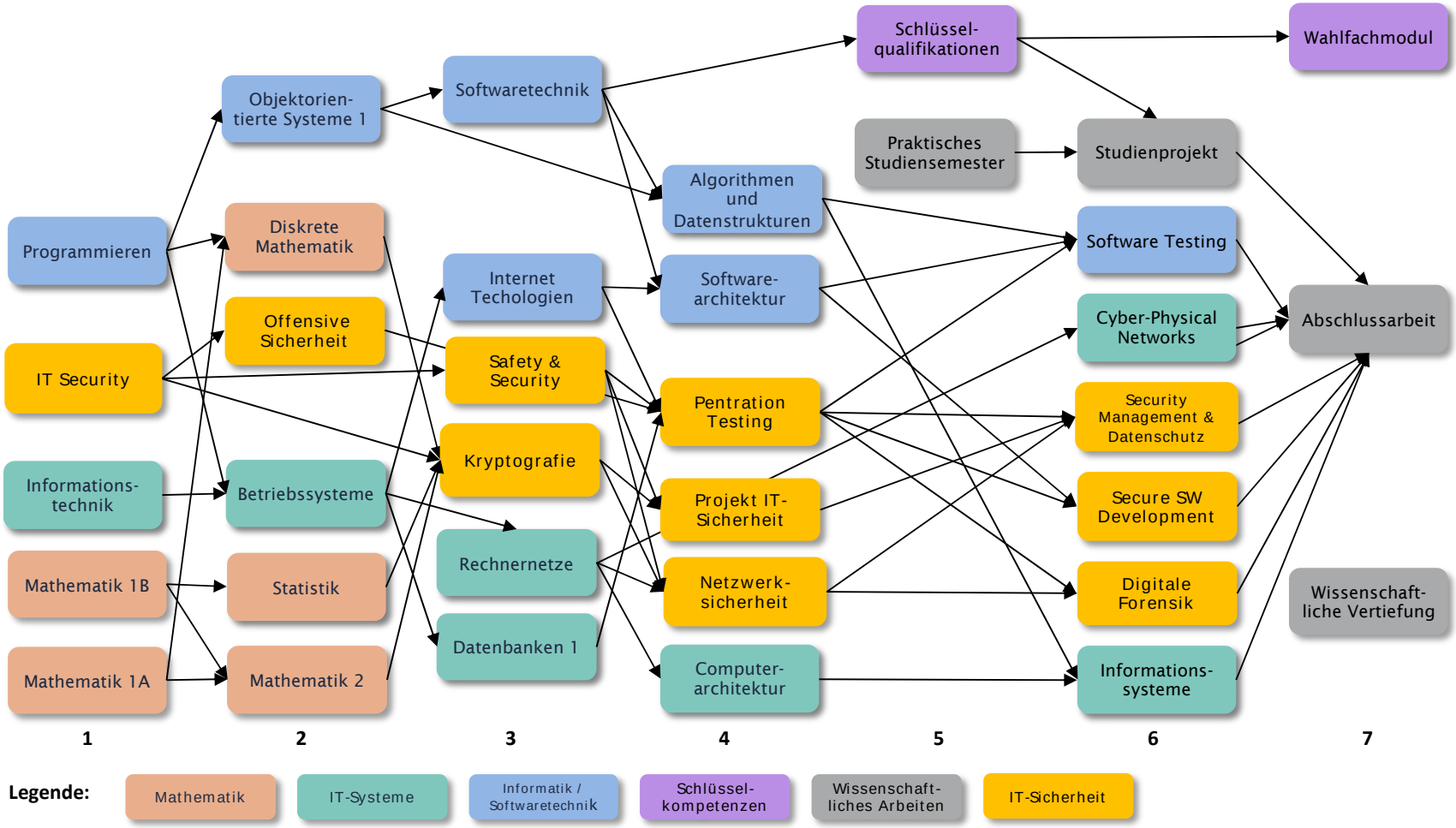
ISB	IT-Sicherheit (Bachelor)
IEP	Ingenieurpädagogik Informationstechnik-Elektrotechnik (Bachelor)
SWB	Software und Medieninformatik (Bachelor)
TIB	Technische Informatik (Bachelor)
WKB	Wirtschaftsinformatik (Bachelor)

Version: 27.7.2022

Übersicht Modulplan IT-Sicherheit

Übersicht Modulabhängigkeiten Studienschwerpunkt IT-Sicherheit - Erreichen des Gesamtziels

Die folgende Grafik zeigt die Verwendung der Module durch andere Module. Ein Pfeil von einem Modul zu einem anderen Modul bedeutet dabei, dass die Inhalte des vorigen Moduls im zweiten Modul verwendet werden oder dass die Inhalte für das Erreichen des Modulziels des zweiten Moduls nützlich sind.



Hinweis: Die Pfeile stellen die Modulverbindungen dar, die zum Erreichen des Gesamtziels beitragen. Verbindungen zwischen Modulen innerhalb eines Semesters wurden zugunsten der Übersichtlichkeit nicht dargestellt.

Modulbeschreibung IT Security

Schlüsselworte: Sicherheitskonzepte, Angreifer und Angriffe, Risiko, Sicherheitsmaßnahmen

Zielgruppe:	**1. Semester ISB**	**Modulnummer:**	**ISB 105 xxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Tobias Heer**		
Dauer des Moduls:	**1 Semester**		
Stand:	**19.05.2022**		

Empfohlene Voraussetzungen:

keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden überblicken die unterschiedlichen Teilbereiche IT-Sicherheit und haben ein tiefes Verständnis der verschiedenen Sicherheitsziele. Sie verstehen den Zusammenhang der IT-Sicherheit mit anderen Bereichen der Informatik und können Faktoren, die zur Steigerung oder Senkung des Sicherheitsniveaus in diesen Bereichen führen, erkennen, einordnen und beschreiben. Die Studierenden kennen verschiedene Angriffsmethoden und sind in der Lage geeignete Gegenmaßnahmen für diese zu identifizieren und umzusetzen.

Die Studierenden können Risiken erkennen, bewerten und geeignete Risikobewältigungsstrategien definieren und umsetzen. Dabei können Sie die Wechselwirkungen zwischen technischen und organisatorischen Sicherheitsmaßnahmen, der wirtschaftlichen Tätigkeit eines Unternehmens und die Wirtschaftlichkeit und Nachhaltigkeit der Umsetzung von verschiedenen Sicherheitsmaßnahmen einordnen und bewerten.

Die Studierenden sind durch ein ganzheitliches Verständnis des Themas IT-Sicherheit in der Lage, weitere Lehrinhalte anderer Module einzuordnen und im Sinne der Sicherheit zu hinterfragen, als auch ihre Auswirkung auf die Sicherheit von Systemen einzuschätzen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Sicherheitsziele und Bedrohungen dieser
- Arten von Angriffen und Angreifertypen
- Sicherheitsmechanismen und deren Wirkweisen

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Die Wirkung verschiedener Sicherheitsmechanismen einschätzen
- Das Sicherheits-Risiko eines Unternehmens systematisch erfassen und bewerten
- Geeignete Sicherheitsmechanismen auswählen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- Klar mit Teammitgliedern über Themen der Sicherheit sprechen zu können und verantwortungsbewusst zu handeln.

Inhalt:

- Terminologie und Konzepte der Informationssicherheit
- Schutzziele und deren Bedrohungen
- Aktuelle Bedrohungslage und Gefahren für die IT-Sicherheit
- Angriffe, Angreifermotivation und Angreifer-Typen

- Sicherheit als Prozess im Unternehmen
- Sicherheitslücken und deren Ausnutzung durch Angriffe
- Sicherheitsorganisation, Sicherheitsinfrastruktur und Sicherheitsrichtlinien, Erkennung, Bewertung und Vermeidung von Sicherheitsrisiken in Unternehmen
- Sicherheitsmaßnahmen technischer und organisatorischer Form
- Schutz der Vertraulichkeit und Integrität durch kryptografische Mechanismen
- Schutz der Verfügbarkeit durch Datensicherung und Datenwiederherstellung

Literaturhinweise:

- C. Eckert: IT-Sicherheit, Oldenbourg Wissenschaftsverlag, München, 2018.
- W. Stalling: Computer Security: Principles and Practice, Pearson Education, 2018.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Informationstechnik

Schlüsselwörter: Methodische Anwendung eines Rechners

Zielgruppe:	**1. Semester SWB, ISB**	**Modulnummer:**	**IT 105 1002**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Reiner Marchthaler**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben ein grundlegendes Verständnis für die Arbeitsweise eines Computers. Sie haben Grundkenntnisse über den grundlegenden Aufbau, die Architektur und die prinzipielle Funktionsweise eines modernen Rechners. Darüber hinaus ist ein Grundverständnis für die Codierung von Zahlen und Zeichen sowie für kombinatorische Logik vorhanden. Sie sind in der Lage, die Besonderheiten verschiedener Betriebssysteme darlegen zu können.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Arbeitsweise eines Computers,
- die Architektur moderner Rechner,
- die Zahlendarstellung in Computern.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Boolesche Algebra anwenden und einfach kombinatorische Schaltungen entwickeln.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- verschiedene Rechnerarchitekturen und die Besonderheiten verschiedener Betriebssysteme zu benennen.

Inhalt:

- Aufgaben und Einsatzgebiete von Rechnern
- Zahlen- und Zeichencodierung (Zahlenbereich, Auflösung, Überläufe)
- Boolesche Algebra und Kombinatorische Schaltungen
- Aufbau und Architektur eines modernen Rechners
- Aufbau einer CPU, Speicher und Ein-/Ausgabe
- Überblick Betriebssysteme und Anwendungsprogramme

Literaturhinweise:

- Gumm, Heinz-Peter und Sommer, Manfred: Einführung in die Informatik, 10. Auflage, Oldenbourg Verlag, 2013.
- Hoffmann, Dirk: Grundlagen der Technischen Informatik, Hanser Verlag, 2013.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mathematik 1A

Schlüsselworte: Funktionen, Differential- und Integralrechnung

Zielgruppe:	**1. Semester SWB, ISB**	**Modulnummer:**	**IT 105 1003**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Jürgen Koch**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Elementarmathematik aus der Schule, insbesondere Kenntnisse über Funktionen

Modulziel – angestrebte Lernergebnisse:

- Die Studierenden werden in die Lage versetzt, mathematische Problemstellungen mit Funktionen analytisch zu lösen

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Eigenschaften von Funktionen in einer und in mehreren Veränderlichen
- anschauliche und mathematische Bedeutung der Begriffe „Grenzwert“, „Stetigkeit“, „Ableitung“ und „Integral“

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- mithilfe von Differential- und Integralrechnung Eigenschaften von Funktionen analytisch zu bestimmen.

Übergreifende Kompetenzen

Die Studierenden können

- Problemstellungen systematisch analysieren und lösen
- logische Schlussfolgerungen nachvollziehen

Inhalt:

- Elementare Funktionen und ihre Eigenschaften
- Folgen, Grenzwerte und Stetigkeit
- Differentialrechnung
- Integralrechnung
- Funktionen mit mehreren Variablen

Literaturhinweise:

- Koch, J und Stämpfle, M.: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017(ISBN 9783446451667).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	5 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mathematik 1B

Schlüsselworte: Vektoren, Matrizen, komplexe Zahlen

Zielgruppe:	**1. Semester WKB, ISB**	**Modulnummer:**	**IT 105 1004**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Jürgen Koch**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Elementarmathematik aus der Schule

Modulziel – angestrebte Lernergebnisse:

- Die Studierenden werden in die Lage versetzt, mathematische Problemstellungen mit Vektoren, Matrizen und komplexen Zahlen analytisch zu lösen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Begriffe und Eigenschaften von Vektoren, Matrizen und komplexen Zahlen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- Berechnungen mit linearen Gleichungssystemen, Vektoren, Matrizen und komplexen Zahlen durchzuführen.

Übergreifende Kompetenzen

Die Studierenden können

- Problemstellungen systematisch zu analysieren und zu lösen
- logische Schlussfolgerungen nachvollziehen

Inhalt:

- Lineare Gleichungssysteme
- Vektoren
- Matrizen
- komplexe Zahlen

Literaturhinweise:

- Koch, J und Stämpfle, M.: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017(ISBN 9783446451667).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	5 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Programmieren

Schlüsselworte: Programmierkonzepte, Algorithmen

Zielgruppe:	**1. Semester SWB, ISB**	**Modulnummer:**	**IT 105 1015**
Arbeitsaufwand:	**10 ECTS**		**300 h**
Davon	**Kontaktzeit**		**240 h**
	Selbststudium		**30 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Mirco Sonntag**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, technische Aufgabenstellungen zu verstehen, einen Algorithmus zur Lösung der Aufgabe zu entwickeln und anschließend auf Basis des Algorithmus ein Programm in einer Programmiersprache zu erstellen.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- atomare Befehle und Kontrollstrukturen einer Programmiersprache
- Variablen und Konstanten
- elementare, abgeleitete und zusammengesetzte Datentypen
- das Prinzip der prozeduralen Programmierung
- ein Werkzeug zur Erstellung von Programmen

Fertigkeiten – methodische Kompetenzen
Die Studierenden sind in der Lage

- von Aufgabenstellungen Algorithmen abzuleiten
- aus diesen Algorithmen selbstständig Programme zu entwickeln
- grundlegende Entscheidungen über den Programmentwurf zu treffen

Übergreifende Kompetenzen
Die Studierenden können

- mit einer integrierten Entwicklungsumgebung Programme erstellen

Inhalt:

- Grundlagen
  - Programmieren
  - Werkzeuge der Programmerstellung
  - Umsetzung von Aufgabenstellungen in Algorithmen
  - Speicherverwaltung, Stack und Heap

- Einführung in eine Programmiersprache
  - Elementare Datentypen, Variablen und Konstanten
  - Abgeleitete und zusammengesetzte Datentypen (Felder, Zeichenketten, Strukturen, Zeiger)
  - Ausdrücke mit Operatoren und Zuweisungen
  - Kontrollstrukturen zur Verzweigung und Iteration
  - Prozedurale Programmierung, call-by-value und call-by-reference
  - Rekursive Funktionen
  - Operationen auf Dateien

Literaturhinweise:

- Dausmann et al.: C als erste Programmiersprache. Vieweg+Teubner, 2010.
- Erlenkötter: C von Anfang an. rororo, 1999.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lernergebnisse:

Die Studierenden beherrschen die Methoden zur Erstellung von Programmen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Betriebssysteme

Schlüsselwörter: **Prozess-/ Speicherverwaltung, IPC, Systemprogrammierung, UNIX**

Zielgruppe:	**2. Semester SWB, ISB**	**Modulnummer:**	**IT 105 2004**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Rainer Keller**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse im Programmieren mit C

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz zur Nutzung von Computer-Hardware und Software sowie von Betriebssystemen und Rechnernetzen. Sie können die grundlegenden Konzepte von Betriebssystemen beschreiben und die in den marktgängigen Betriebssystemen realisierten Lösungen bewerten.

Kenntnisse – fachliche Kompetenzen

Sie kennen

- die wesentlichen Funktionen und Dienste von Betriebssystemen und sind in der Lage, sie interaktiv oder in Anwendungsprogrammen zu nutzen.
- die Mechanismen der Authentisierung und Autorisierung.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind können

- den Zugriff von NutzerInnen auf Computer, Dienste und Daten angemessen regeln.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- die grundlegenden Konzepte von Betriebssystemen zu beschreiben und die in den marktgängigen Betriebssystemen realisierten Lösungen zu bewerten.

Inhalt:

- Einführung in die Aufgaben und die Struktur von Betriebssystemen
- Benutzung von UNIX und Windows per Kommandozeile (Shell- / Skript-Programmierung)
- Prozesse und Threads
- Linux Kernel Module
- Speicherverwaltung
- Interprozesskommunikation und Synchronisation
- Dateisysteme
- Input und Output
- Security
- Container, Virtualisierung und Cloud

Literaturhinweise:

Tanenbaum, A.S.: Moderne Betriebssysteme, 4. Akt. Auflage, Pearson 2016.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Nachbereitung und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden sind in der Lage, ein vernetztes UNIX-System sowohl von der Kommandozeile, als auch von einer grafischen Benutzungsoberfläche aus zu bedienen und häufig wiederkehrende Aufgaben durch Shell-Skripte zu automatisieren. Sie beherrschen die Programmierung von Anwendungen, die die Funktionen und Dienste des Betriebssystems durch POSIX-konforme Programmierschnittstellen nutzen. Die Studierenden sind befähigt, die wichtigsten Netzwerkdienste von Betriebssystemen Client-seitig zu nutzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Diskrete Mathematik

Schlüsselworte: Zahlentheorie, Algebra

Zielgruppe:	**2. Semester SWB, ISB**	**Modulnummer:**	**SWB 105 2024**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Karin Melzer**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnis von linearen Gleichungssystemen, Vektoren, Matrizen, Funktionen in einer und in mehreren reellen Veränderlichen, komplexe Zahlen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden können konkrete Anwendungen in der Informatik durch abstrakte mathematische Methoden analysieren und lösen. Sie werden in die Lage versetzt, mathematische Problemstellungen der Theoretischen Informatik und der Kryptografie mathematisch zu lösen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Beweistechniken
- Mengen und Relationen
- Begriffe und Sätze der elementaren Zahlentheorie
- grundlegende algebraische Strukturen und ihre Eigenschaften

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Teilbarkeits-, Modulo- und Kongruenzberechnungen mit ganzen Zahlen und algebraischen Strukturen durchführen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- logische Schlussfolgerungen nachzuvollziehen
- konkrete Anwendungen durch abstrakte mathematische Methoden zu analysieren und zu lösen

Inhalt:

- Beweistechniken, vollständige Induktion, Aussagenlogik
- Mengenlehre und Relationen,
- Zahlentheorie: Teilbarkeit, Module, Kongruenz, Arithmetik, Division mit Rest, multiplikative Inverse, Primzahlen, Euklidischer Algorithmus, Kleiner Satz von Fermat, Großer Satz von Fermat, Eulersche Funktion, Diophantische Gleichungen, Chinesischer Restsatz,
- Algebraische Strukturen und Unterstrukturen: Monoide, Gruppen, Ringe, Körper, Ordnung von Elementen, zyklische Gruppen, Generatoren, Vektorräume
- Polynomringe und Galois Körper, Faltung
- Anwendungsbeispiele aus dem Bereich der symmetrischen und asymmetrischen Verschlüsslung, sowie Protokollen der Rechnerkommunikation werden exemplarisch behandelt

Literaturhinweise:

- Koch, J und Stämpfle, M.: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017(ISBN 9783446451667).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mathematik 2

Schlüsselworte: Differentialgleichungen, Differenzengleichungen, Potenzreihen, Fourier-Reihen

Zielgruppe:	**2. Semester SWB, ISB**	**Modulnummer:**	**SWB 105 2003**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Jürgen Koch**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Lineare Gleichungssysteme, Vektoren, Matrizen, Funktionen in einer und in mehreren reellen Veränderlichen, komplexe Zahlen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, naturwissenschaftliche und technische Problemstellungen mathematisch zu lösen.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die wichtigsten Begriffe und Eigenschaften von Differentialgleichungen, Differenzengleichungen, Potenzreihen und Fourier-Reihen

Fertigkeiten – methodische Kompetenzen
Die Studierenden können:

- Differentialgleichungen und Differenzengleichungen lösen
- Funktionen als Potenzreihen darstellen
- Periodische Funktionen durch Fourier-Reihen analysieren

Übergreifende Kompetenzen
Die Studierenden sind in der Lage,

- Problemstellungen systematisch zu analysieren und zu lösen
- logische Schlussfolgerungen nachzuvollziehen

Inhalt:

- Lineare und nichtlineare Differentialgleichungen
- Lineare Differentialgleichungssysteme
- Lineare Differenzengleichungen und Differenzengleichungssysteme
- Potenzreihen und Taylor-Reihen
- Fourier-Reihen

Literaturhinweise:

- Koch, J und Stämpfle, M.: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017(ISBN 9783446451667).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können Problemstellungen aus Naturwissenschaft und Technik mithilfe mathematischer Modelle am Computer lösen, simulieren und visualisieren.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Offensive Sicherheit

Schlüsselwörter: Schwachstellen, Penetration Testing, Angriffswerkzeuge

Zielgruppe:	**2. Semester ISB**	**Modulnummer:**	**ISB 105 XXXX**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Projektarbeit		**70 h**
	Ausarbeitung Referat		**20 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Tobias Heer**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

IT-Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz, einfache technische Sicherheitsüberprüfungen und Sicherheitsbewertungen durchzuführen und offensive Methoden anzuwenden. Die Studierenden entwickeln dabei eine Vorstellung des moralisch-rechtlichen Rahmens und der Verantwortung bei offensiven Sicherheitstests. Die Studierenden üben sich in der eigenständigen Erarbeitung offensiver Konzepte und Werkzeuge.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- Ziele und Grenzen von offensiven Sicherheitstests
- Methodik und Ablauf eines Penetrationstest
- praktische, zwischenmenschliche und rechtliche Grenzen
- die moralischen und ethischen Grenzen beim Einsatz offensiver Methoden

Fertigkeiten – methodische Kompetenzen
Die Studierenden können

- die vorbereitenden Schritte eines Penetrationstests anwenden
- können ausgewählte offensive Techniken anwenden
- einfache technische Sicherheitsanalysen mit offensiven Werkzeugen verantwortungsvoll durchführen

Übergreifende Kompetenzen
Die Studierenden sind in der Lage,

- technische Risiken beim Einsatz von Testwerkzeugen zu überblicken und zu bewerten
- ein komplexes Thema in Kleingruppen zu bearbeiten und zu präsentieren

Inhalt:

- Motivation und Entstehung offensiver Sicherheitsmethoden
- Rechtliche und moralische Grundlagen
- Ablauf von Penetrationstests (Testvorbereitung, Informationsbeschaffung, Zielanalyse, Angriff, Dokumentation und Abschlussgespräch)
- Vorgehen und Grenzen bei individuellen Sicherheitstests (z.B. Bug Bounty)
- Kleinprojekte zu offensiven Methoden und Werkzeugen z.B. Webbasierte Angriffe, Buffer-Overflows, Scanning and Enumeration, Exploits, Angriffe gegen verschiedene Betriebssysteme, Social Engineering, Physische Sicherheit, etc.

Literaturhinweise:

- Institute for Security and Ipen Methodologies, Open Source Security Testing Methodology Manual (OSSTM), online: www.isecom.org/osstmm/
- Hadnagy, C: Social Engineering. The Art of Human Hacking
- Scarfone, K. A., Souppaya, M. P., Cody, A., & Orebaugh, A. D.: Sp 800-115. technical guide to information security testing and assessment.
- Pentest-standard.org. (2018). The Penetration Testing Execution Standard. http://www.pentest-standard.org/index.php/Main_Page.
- Wechselnde Online-Literatur für die Präsentation des Projekts

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Einführende Vorlesung und Seminar
Leistungskontrolle:	Projekt und Referat (20 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernziele:

Die Studierenden sind in der Lage, einfache Sicherheitsüberprüfungen mithilfe von offensiven Sicherheitsmethoden durchzuführen.

Bildung der Modulnote:

Projekt mit abschließendem Referat (20 Minuten)

Modulbeschreibung Objektorientierte Systeme 1

Schlüsselwörter: Objektorientierte Programmierkonzepte

Zielgruppe:	**2. Semester SWB, ISB**	**Modulnummer:**	**IT 105 2027**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Andreas Rößler**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse einer Programmiersprache

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben eine fundierte Grundlagenausbildung in Informatik und Programmieren. Sie beherrschen die Programmiersprache C++.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- Klassenkonzepte
- Operatoren und Overloading
- Vererbung und Polymorphie

Fertigkeiten – methodische Kompetenzen
Die Studierenden können

- abstrakte Klassen erstellen
- Programme in C++ erstellen

Übergreifende Kompetenzen
Die Studierenden sind in der Lage,

- Programme in C++ methodisch zu programmieren

Inhalt:

Es werden grundlegende Konzepte der objektorientierten Programmierung vermittelt.
Hierzu gehören:

- Klassenkonzept (Attribute, Methoden), Information-Hiding (public, private),
- Konstruktoren und Destruktoren
- Statische Variablen und statische Methoden
- Operatoren und Overloading
- Vererbung und Polymorphie
- Abstrakte Klassen und ihre Rolle als Schnittstellendefinition

Als weitere Themen, die bei der objektorientierten Software-Entwicklung wichtig sind, werden behandelt:

- Referenzen, Namensräume, Umgang mit Strings
- Definition und Behandlung von Ausnahmen
- Bearbeitung von Dateien mit Hilfe von Streams
- Cast-Operatoren und die Typbestimmung zur Laufzeit

Literaturhinweise:

- Stroustrup, Bjarne: Einführung in C++, Pearson Verlag, 2010 (ISBN 9783868940053).
- Wolf, Jürgen: C++, Galileo Computing, 2014 (ISBN 978-3-8362-3895-3).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden beherrschen die methodische Programmierung objektorientierter Systeme.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Statistik

Schlüsselwörter: Kombinatorik, Wahrscheinlichkeitsrechnung, Statistik

Zielgruppe:	**2. Semester SWB, ISB**	**Modulnummer:**	**IT 105 2018**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Gabriele Gühring**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Funktionen in einer und in mehreren reellen Veränderlichen, Matrizenrechnung

Gesamtziel:

Die Studierenden werden in die Lage versetzt, zufällige und mit Unsicherheit behaftete Phänomene zu beschreiben, zu erklären und zu verstehen. Sie beherrschen die grundlegenden Methoden der Wahrscheinlichkeitsrechnung, Statistik und Kombinatorik.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die grundlegenden kombinatorischen Formeln und ihre Anwendbarkeit auf entsprechende Fragestellungen,
- die grundlegenden wahrscheinlichkeitstheoretischen Kennzahlen und ihre Berechnungen bzw. Beziehungen untereinander,
- die grundlegenden statistischen, diskreten und stetigen Verteilungen
- die Grundlagen der beschreibenden Statistik,als auch der schließenden Statistik und können sie auf spezifische Situationen anwenden.

Fertigkeiten – methodische Kompetenzen
Die Studierenden sind in der Lage,

- große Datensätze zu beschreiben und Informationen darzustellen
- Ereignisse mit Häufigkeiten, Mittelwert und Varianz bzw. Standardabweichung zu beschreiben
- Aussagen über mit Unsicherheit behaftete Probleme zu bewerten und einzuordnen

Übergreifende Kompetenzen
Die Studierenden können

- Aussagen über mit Unsicherheit behaftete Fragestellungen herleiten, bewerten, einordnen
- Statistik als wichtiges Instrument zur Unterstützung der Arbeit mit großen Datenmengen

Inhalt:

- Datengewinnung und Datenbereinigung
- Darstellung statistischen Materials (Merkmaltypen, grafische Darstellung, Lageparameter einer Stichprobe)
- Mehrdimensionale Stichproben (Korrelation und Regression)
- Kombinatorik
- Wahrscheinlichkeitsrechnung (Laplace-Modelle; Zufallsvariablen und Verteilungsfunktionen; spezielle Verteilungsfunktionen wie z. B. Normal- oder Binomialverteilung)
- Schließende Statistik, insbesondere statistische Testverfahren und Vertrauensbereiche, p-Wert
- Einführung in stochastische Prozesse

Literaturhinweise:

- Sachs, L.: Angewandte Statistik: Methodensammlung mit R, Springer Verlag, 16. Auflage 2018 (ISBN 3662566567).
- Ross, S.: Statistik für Ingenieure und Naturwissenschaftler, 3. Auflage, Spektrum Verlag, 2006 (ISBN 3827416213).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden beherrschen eine Anwendungssoftware, mit der sie statistische Fragestellungen auswerten und darstellen können.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Datenbanken 1

Schlüsselwörter: SQL, ODBC, Transaktionen, DBMS-Administration

Zielgruppe:	**3. Semester SWB, ISB**	**Modulnummer:**	**IT 105 3007**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch und Englisch**		
Modulverantwortung:	**Prof. Jürgen Nonnast**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse in Betriebssystemen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erlernen die Grundkonzepte von hierarchischen, netzwerkorientierten, relationalen und objektorientierten Datenmodellen. Sie sind in der Lage, Datenbank-anwendungen zu entwickeln. Sie können Datenbank-Anwendungen nach Vorgaben entwickeln. Sie beherrschen die Konzepte der Funktionsweise und des Betriebs von Datenbank-Managementsystemen und können diese bewerten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Algebraische Relationen
- SQL-Funktionen

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- Embedded SQL mit C anwenden,
- Verknüpfungen von Tabellen erstellen
- DML- und DDL-Zugriffe durchführen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- ein Datenbank-Managementsystemen zu konfigurieren.

Inhalt:

- Grundlagen von Datenmodellen
- Relationen Algebra
- SQL: Projektion, Restriktion, Unterabfragen, Skalare Funktionen, Aggregatfunktionen
- Datumsfunktionen
- DML-Zugriffe und DDL-Zugriffe
- Verknüpfung von Tabellen (Inner, Left, Right, Outer Join)
- Embedded SQL mit C (Singleton Select, Cursor Select, Cursor Update)
- Betrachtungen zur portablen Applikationsentwicklung mit SQL99
- Aufbau und Funktionsweise eines Datenbank-Managementsystems mit besonderem Fokus auf Mehrbenutzerbetrieb und Performance, Datensicherheit, Verfügbarkeit

Literaturhinweise:

- Baklarz, Zikopoulos: DB2 9 DBA Guide, Reference, and Exam Prep, IBM Press, 2007.
- Sanders, E.: DB2 9 Fundamentals: Certification Study Guide, MC Press Online, 2007.
- Sanders, E.: DB2 9 Database Administration: Certification Study Guide MC Press Online, 2007.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden können Betriebskonzepte nach Vorgabe realisieren.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Internet Technology

Keywords:	**Internet, Web, Client, Server, HTTP, HTML, CSS, Javascript, PHP**

Audience:	**4. Semester SWB, ISB**	**Module number:**	**IT 105 3010**
Workload:	**5 ECTS**		**150 h**
divided into	**Contact time**		**90 h**
	Self-study		**30 h**
	Exam preparation		**30 h**
Course language:	**English**		
Modul director:	**Prof. Dr. -Ing. Harald Melcher**		
duration of the module:	**one semester**		
Vaild from:	**01.09.2019**		

Recommended requirements:

Knowledge in an object oriented programming language like Java or C#. Routine in a development IDE like IntelliJ or VisualStudioCode.

Desired learning outcomes of the module:

Students are proficient in selecting the right tools for Web based client server applications. They know the security risks and how to mitigate them and they have a basic understanding of the programming languages in use for Web applications.

Knowledge - professional competences

- Students acquire knowledge in the area of web based applications and services. They gain an overview over the protocols, the interworking of clients and servers and the major languages of the Internet.

Skills - methodical competences

- Students are able to appraise the best combination of technologies for a specific web task. They can estimate the risk of a given solution.

Comprehensive Competencies

- Students understand how web based services interact and are able to self develop a simple service.

Contents:

- Basic structure of client – server communication
- Basic functions of a web server
- The web protocol HTTP
- Use of markup languages like HTML or XML
- Design and implementation of interactive web applications with HTML, CSS, Javascript and JSON

Literature:

- Freeman & Robson, Head First HTML5 Programming, O'Reilly
- Freeman & Robson, Head First HTML and CSS, O'Reilly
- Crockford, Javascript: The good Parts, O'Reilly
- Chaffer & Swedberg, Learning jQuery, Packt Publishing
- Bibeault & Katz, jQuery in Action, Manning

Offered:

Each semester

Sumodules and Assessment:

Type of instruction:	Lecture with exercises and exam preparation
Type of assessment:	Exam (90 minutes)
Hours per week:	3 SWS
Estimated student workload:	120 hours
Type of instruction:	Lab Work
Type of assessment:	Report and presentation
Hours per week:	1 SWS
Estimated student workload:	30 hours

Learning outcomes:

Students are proficient in developing simple Web Applications according to best practice examples. They have experienced the pitfalls of Javascript and CSS programming and know how to cope with them.

Generation of the module grade:

Exam graded, report and presentation ungraded

Modulbeschreibung Kryptografie

Schlüsselwörter: Verschlüsselung, Chiffren, Angriffe

Zielgruppe:	**3. Semester ISB**	**Modulnummer:**	**ISB 105 XXXX**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Tobias Heer**		
Dauer des Moduls:	**ein Semester**		
Stand:	**20.5.2022**		

Empfohlene Voraussetzungen:

Grundverständnis von modularer Arithmetik, Grundverständnis der Wahrscheinlichkeitsrechnung, Grundlagen zur IT-Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden verstehen, wie kryptografische Verfahren funktionieren, kennen deren Limitierungen, können diese auswählen und sicher einsetzen. Neben mathematischen Grundlagen liegt der Fokus der Vorlesung auf den in der Praxis anwendbaren Chiffren und Mechanismen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die mathematischen Grundlagen gängiger kryptografischer Systeme
- die historische Entwicklung der symmetrischen Kryptografie
- symmetrische und asymmetrische Verschlüsselungsverfahren
- Verfahren zum Schutz der Integrität
- Verfahren zur Authentifizierung
- Hash-Funktionen und deren Limitierungen
- Verfahren zum Schlüsselaustausch

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- Kryptographische Methoden einordnen und deren inhärenten Stärken und Schwächen verstehen
- Angriffe gegen moderne und historische Chiffren verstehen und anwenden
- Die Stärke aktueller kryptografischer Verfahren einschätzen und anwendungsbezogen sinnvolle Verfahren auswählen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- die Entwicklung und Anwendung von Verschlüsselungsmechanismen im Kontext von konkreten Anwendungen einzuordnen und zu verstehen
- Bezüge zwischen den Schutzzielen und den einsetzbaren kryptografischen Mechanismen herzustellen
- Realistische Annahmen über die Sicherheit und die Limitierungen von Systemen mit kryptografischen Mechanismen zu machen

Inhalt:

- Grundlagen der symmetrischen Verschlüsslung
  - Substitution und Transposition
  - Cäsar-, Skytale-, Homomphone-, Vigenère-, und Vernam-Verschlüsselung sowie One-Time-Pad

    - Angriffe gegen symmetrische Chiffren
- Moderne symmetrische Verschlüsslungsverfahren
    - Herleitung: Data Encryption Standard (DES)
    - Advanced Encryption Standard (AES)
    - Schwächen und Angriffe
- Zufallszahlen und Zufallszahlengeneratoren
    - Entropie und Zufälligkeit
    - Zufallsquellen und Pseudozufallszahlengeneratoren
    - Schlüssel und Schlüssellängen
- Hash-Funktionen
    - Merkle Damgård Verfahren
    - SHA-Familie (SHA-1, SHA-2)
    - Schwamm-Konstruktion und SHA-3
    - Stärke und Angriffe auf Hash-Funktionen
- Angewandte asymmetrische Kryptografie
    - Schlüsselaustausch mit RSA und Diffie Hellman
    - Schlüsselaustauschverfahren auf elliptischen Kurven
- Digitale Signaturen und Public-Key Infrastruktur
    - Signatur-Algorithmen und deren Anwendungen
    - Digitale Zertifikate und deren Lebenszyklus
- Überblick: Quantencomputer und Post-Quanten Kryptografie
    - Quantencomputer und Kryptografie
    - Shors Algorithmus
    - Grovers Algorithmus
    - Post-Quanten-Systeme und deren Limitierungen

Literaturhinweise:

- Schmeh, K.: Kryptografie. dpunkt.verlag, 2016 (ISBN 978-3-86490-356-4).
- Beutelspacher Albrecht, Schwenk, Jörg, Wolfenstetter, Klaus-Dieter: Moderne Verfahren der Kryptographie – Von RSA zu Zero-Knowledge. 8. Auflage, Springer Spektrum, 2015 (ISBN 978-3-8348-1927-7 (Papier); 10.1007/978-3-8348-2322-9 (DOI)).
- Buchmann, J.: Einführung in die Kryptographie. 6. Auflage, Springer Spektrum, 2016 (ISBN 978-3-642-39774-5 (Papier); 10.1007/978-3-642-39775-2 (DOI)).
- Schneier, B.: Applied Cryptography. Protocols, Algorithms, and Source Code in C. Wiley, New York 1996.
- Wechselnde aktuelle Online-Literatur und Standards

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Studierende haben die Fähigkeit, kryptografische Mechanismen richtig auszuwählen und sicher einzusetzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Softwaretechnik

Schlüsselworte: Software Engineering, Modellierung, Qualitätssicherung

Zielgruppe:	**3. Semester ISB**	**Modulnummer:**	**IT 105 3039**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr. Mirko Sonntag**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse einer höheren Programmiersprache

Modulziel – angestrebte Lernergebnisse:

Die Studierenden verstehen und beherrschen ingenieursmäßiges Software-Engineering. Die Studierenden verfügen über Wissen in den Bereichen ingenieursmäßige Software-Entwicklung, Vorgehensmodelle, Anforderungsanalyse, Qualitätssicherung, Modellierung und Versionsverwaltung.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Die Notwendigkeit für ingenieursmäßige Software-Entwicklung
- Plangetriebene und agile Vorgehensmodelle zur Software-Entwicklung
  - Phasen, Meilensteine und Artefakte
  - Rollen und Aufgaben
- Methoden zum Aufnehmen von Anforderungen
- Software-Spezifikation und -Entwurf
- Maßnahmen zur Sicherung der Softwarequalität
- Versionsverwaltung und Konfigurationsmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- zwischen einem plangetriebenen oder agilen Vorgehensmodell zu entscheiden
- planvoll Anforderungen aufzunehmen und zu dokumentieren
- eine Software-Spezifikation und einen Software-Entwurf zu erstellen
- IT-Projekte durchzuführen, die eine hohe Software-Qualität sicherstellen
- mit einer Versionsverwaltung umzugehen

Übergreifende Kompetenzen

Die Studierenden können

- Methoden des Software Engineering anwenden und damit ein IT-Projekt durchführen

Inhalt:

- Prinzipien des Software Engineering
- Plangetriebene und agile Vorgehens- und Prozessmodelle
- Requirements Engineering
- Systemspezifikation
- Systementwurf
- UML
  - Modellelemente: Knoten, Kanten, Beschriftungen
  - Beziehungen: Assoziation, Multiplizität, Qualifizierung, Generalisierung, Aggregation und Komposition
  - Use Case-, Klassen-, Objekt-, Sequenz-, Aktivitäts- und Zustandsdiagramme
- Versionsverwaltung und Konfigurationsmanagement
- Software-Qualität, Einführung in Software-Testing
- Software-Projektmanagement

Literaturhinweise:

- Ludewig & Lichter: Software Engineering, dpunkt, 2007.,.
- Sommerville: Software Engineering, 2011, Addison-Wesley.
- Brügge & Dutoit: Object-Oriented Software Engineering, 3rd edition, 2010, Prentice Hall.
- Baumgartner et al.: Agile Testing, 2018, Hanser.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden beherrschen die Methoden der agile Software-Entwicklung, des Requirements Engineering, der Modellierung mit UML, des Unit-Testing und der Versionsverwaltung.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Blockseminar Software-Projekt Management
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden erlernen das erfolgreiche Durchführen von Projekten. Sie beherrschen die Instrumente des Projektmanagements.

Bildung der Modulnote:

Klausur, zwei unbenotete Testate

Modulbeschreibung Safety and Security

Schlüsselworte: Sicherheit und Risiko, Risikoanalyse, Risikomanagement, Funktionssicherheit von E/E/PE-Systemen

Zielgruppe:	**3. Semester SWB, ISB, TIB**	**Modulnummer:**	**TIB 105 6027**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr. Dominik Schoop**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

- Grundlagen IT-Sicherheit
- Grundlagen in Mathematik, Statistik und Stochastik
- Grundlagen in Physik, Elektrotechnik und Softwareentwicklung
- Kenntnisse zu Rechnernetzen und Computerarchitekturen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben Kenntnisse über grundlegende Konzepte und Vorgehensweisen der Sicherheitstechnik mit Bezug zur IT-Sicherheit.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Begriffe der Sicherheitstechnik und der IT-Sicherheit,
- Vorgehensweise und Ziele der Risikoanalyse,
- Grundlegende Konzepte der Sicherheitstechnik.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- Risikobetrachtungen zu verstehen und zu erstellen,
- Sicherheitsziele für E/E/PE-Systeme zu definieren,
- sicherheitsgerichtete Konzepte zu bewerten.

Übergreifende Kompetenzen

Die Studierenden können

- Maßnahmen zur Erhöhung der Sicherheit verstehen und bewerten.

Inhalt:

- Definition und Unterscheidung, sowie Überschneidungen der Begriffe „Funktionssicherheit“ (Safety) und „IT-Sicherheit“ (Security)

- Sicherheitsziele (Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität)
- (kryptographische) Sicherheitsmaßnahmen
- Bedrohungs- und Risikoanalysen der IT-Sicherheit

- Ziele der Funktionssicherheit (Zuverlässigkeit, Verfügbarkeit und Sicherheit
- Risikoanalysen (Risiko-Graph, Fehlerbaumanalyse (FTA))
- Maßnahmen zur Erhöhung des Safety Integrity Levels
- Konzepte für Systeme und Funktionen der Klassen Fail-Safe und Fail-Operational

Literaturhinweise:

- Börcsök, J.: Funktionale Sicherheit – Grundzüge sicherheitstechnischer Systeme
- Ross, H.-L.: Functional Safety for Road Vehicles. Springer Verlag, 2016.
- IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems.
- ISO 26266: Road vehicles – Functional safety, 2018.
- Eckert, C.: IT-Sicherheit, Oldenbourg Wissenschaftsverlag, München, 2018.
- Stalling, W.: Computer Security: Principles and Practice, Pearson Education, 2018.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden beherrschen die Grundlagen der Funktionalen Sicherheit und den Bezug zur IT-Sicherheit.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Rechnernetze

Schlüsselworte: Netztechnik, Protokolle, Ethernet, TCP/IP

Zielgruppe:	**3. Semester SWB, ISB**	**Modulnummer:**	**IT 105 3008**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Michael Scharf**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kompetenzen in den Bereichen Programmierung und Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben Kenntnisse über grundlegenden Konzepte und Technologien in Rechnernetzen. Sie können die grundlegenden Konzepte von Rechnernetzen beschreiben. Sie verstehen das Schichtmodell in Kommunikationsnetzen und die Grundmechanismen und Aufgaben von Kommunikationsprotokollen. Die Funktionsweise wichtiger Standards wie Ethernet und TCP/IP sind den Studierenden bekannt. Dies ermöglicht es ihnen, geeignete Lösungen für verschiedene Anwendungszwecke auszuwählen und zu bewerten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- den Aufbau von Kommunikationsnetzen und das Schichtenmodell,
- die Grundmechanismen und Aufgaben von Protokollen,
- die prinzipielle Arbeitsweise wichtiger Standards wie Ethernet und TCP/IP,
- die Funktionen, Komponenten und Dienste moderner Rechnernetze.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- Kommunikationsdienste zu konfigurieren,
- bestehende Netztechnik und Protokolle zu analysieren,
- Kommunikationsmechanismen gezielt und sinnvoll einzusetzen.

Übergreifende Kompetenzen

Die Studierenden können

- das Zusammenspiel von Rechnernetzen, Betriebssystemen und Anwendungen beschreiben.

Inhalt:

- Grundlagen und Netzarchitekturen
- Kommunikation in lokalen Netzen
- Paketvermittlung im Internet
- Transportprotokolle im Internet
- Internet-Anwendungen
- Technologien in lokalen Netzen
- Technologien in Weitverkehrsnetzen

Literaturhinweise:

- Tanenbaum, Wetherall: Computernetzwerke, Pearson, 2012.
- Kurose, Ross: Computernetzwerke: Der Top-Down-Ansatz, Pearson, 2014.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübungen
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können Netzwerkdienste konfigurieren, Kommunikationsprotokolle nutzen und deren Funktion analysieren und gegebenenfalls Fehler finden.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Netzwerksicherheit

Schlüsselworte: Sichere Kommunikationsprotokolle, Firewalls, Intrusion Detection

Zielgruppe:	**4. Semester ISB**	**Modulnummer:**	**ISB 105 XXX**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Tobias Heer**		
Dauer des Moduls:	**ein Semester**		
Stand:	**20.05.2022**		

Empfohlene Voraussetzungen:

- Kryptografie
- Rechnernetze
- IT-Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden verstehen, wie Netzwerke mit grundlegenden und fortgeschrittenen Sicherheitsmethoden geschützt werden.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Netzwerksicherheitsziele und grundlegende Angriffe
- Sicherheitsmechanismen auf verschiedenen Netzwerkebenen (z.B. PPP, IPsec, TLS, SSH)
- Authentifizierungsframeworks und Identitätsverwaltung (z.B. OAuth2, Kerberos undRADIUS)
- Grundlegende Schutzlösungen und Geräte (z.B. Firewalls, VLAN, VPN, Netzwerküberwachung)
- Erweiterte Sicherheitsmechanismen und -algorithmen (z. B. Intrusion Detection und SIEM)

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- Netzwerksicherheitsmechanismen sinnvoll auswählen und anwenden
- Netzwerke und vernetzte Anwendungen absichern
- Netzwerke in Sicherheitszonen segmentieren
- Netzwerksicherheitsgeräte verstehen und verwenden

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- Systeme mit Kommunikationsmöglichkeiten aufgrund ihrer Gefährdung richtig einzuschätzen und Kommunikationssicherheitsmechanismen beim Entwurf von Systemen zu berücksichtigen.

Inhalt:

- Netzwerksicherheitsziele, Angriffe und Schutzmechanismen
- Sicherheitsmechanismen im Internet und in lokalen Netzen (z. B. VLAN, IEEE 802.1X, IPsec, OpenVPN, TLS, SSH, Kerberos, DKIM, SPF)

- Design und Funktionen von Netzwerksicherheitsprotokollen
- Netzwerkangriffe und Gegenmaßnahmen (z. B. Firewalls, Intrusion Detection-Systeme)
- Sicherer Netzwerkbetrieb und Netzwerküberwachung

Literaturhinweise:

- Stallings, W.: Network Security Essentials, Pearson Prentice Hall, 2007.
- Ferguson, N. & Schneier, B.: Practical Cryptography, John Wiley & Sons, 2003.
- Schäfer, G. & Roßberg, M.: Netzsicherheit, 2. Auflage, dpunkt Verlag, 2014.
- Anderson, R.: Security Engineering, Wiley, 2009.
- Schneier, B.: Applied Cryptography. Protocols, Algorithms, and Source Code in C. Wiley, New York 1996.
- Zahlreiche online verfügbare IETF Internet Standards

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können kryptografische Methoden in der Praxis anwenden.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Computerarchitektur

Schlüsselworte:	**Rechnerarchitektur, Mikroprozessor, Mikrocontroller, Instruction Set Architecture, Assemblerprogrammierung**		
Zielgruppe:	**4. Semester SWB, ISB**	**Modulnummer:**	**IT 105 4003**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr.-Ing. Werner Zimmermann**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Programmieren, Digitaltechnik, Softwaretechnik, Informationstechnik, Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden verstehen den Aufbau und die Funktionsweise von Mikroprozessoren, sowie ihre Peripheriebausteine und können diese programmieren.Sie beherrschen ein Grundverständnis für die Instruction Set Architecture von Rechnern und verstehen, wie die Programmierkonstrukte höherer Programmiersprachen auf die "Sprache der Hardware" abzubilden sind. Sie haben ein Bewusstsein fürdas Zusammenwirken von Programmiersprache, Betriebssystem und Hardware, um effizientere Software zu entwickeln.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Architektur von Rechnersystemen mit Mikroprozessoren und Mikrocontrollern
- die Maschinenbefehlsebene (Instruction Set Architecture) von Rechnern

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Programmierkonstrukte höherer Programmiersprachen auf die "Sprache der Hardware" abbilden,
- Programme in Assembler erstellen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- das Zusammenwirken von Programmiersprache, Betriebssystem und Hardware abzubilden.

Inhalt:

- Aufbau von Rechnersystemen, arithmetisch-logische Operationen, Grundaufgaben von Betriebssystemen
- Programmiermodell (Registersatz, Adressierungsarten, Memory Map, Befehlssatz) eines beispielhaften Mikroprozessors
- Einführung in die Maschinensprache, Abbildung wichtiger Hochsprachenkonstrukte auf die Maschinensprache, Abschätzung des Speicherplatzbedarfs und der Ausführungsgeschwindigkeit
- Hardware/Softwareschnittstelle für typische Peripheriebausteine, digitale und analoge Ein-/Ausgabe, Timer, einfache Netzwerkschnittstellen
- Modulare Programmierung, Schnittstellen für das Zusammenspiel verschiedener Programmiersprachen
- Unterstützung von Betriebssystem-Mechanismen, z.B. Speicherschutz, virtueller Speicher, durch Mikroprozessoren
- Überblick über aktuelle Mikro- und Signalprozessorarchitekturen: Technik und Marktbedeutung

Literaturhinweise:

- Patterson, D.; Hennesey, J.: Computer Architecture and Design. Morgan Kaufmann Verlag, 2008.
- Tanenbaum, A.: Structured Computer Organization. Prentice Hall Verlag, 2012.
- Huang, H.W.: The HCS12/9S12. An Introduction to the hardware and software interface. Thomson Learning Verlag, 2009.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können die Grundlagen der hardwarenahen Programmierung in C/C++ und Maschinensprache (Assembler) in praktischen Übungen umsetzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Penetration Testing

Schlüsselworte: IT-Sicherheit, Pentesting, Offensive Sicherheit, Sicherheitsbewertung

Zielgruppe:	**4. Semester ISB**	**Modulnummer:**	**ISB 105 XXXX**

Arbeitsaufwand:	**5 ECTS**	**150 h**
Davon	**Kontaktzeit**	**60 h**
	Selbststudium	**60 h**
	Prüfungsvorbereitung	**30 h**
Unterrichtssprache:	**Deutsch**	
Modulverantwortung:	**Prof. Dr. Tobias Heer**	
Dauer des Moduls:	**ein Semester**	
Stand:	**01.09.2019**	

Empfohlene Voraussetzungen:

IT Security, Offensive Sicherheit, Safety & Security

Modulziel – angestrebte Lernergebnisse:

Um IT-Systeme erfolgreich gegen unbefugten Zugriff schützen zu können, ist ein Einblick in die Denkweise und Techniken von Angreifern unverzichtbar. Das Modul vertieft das Wissen und die Fertigkeiten der Studierenden bezüglich der die offensive Seite der IT-Sicherheit und behandelt typische Schwachstellen und Angriffsmethoden. Die Studierenden haben einen Überblick über die Vorgehensweise bei Angriffen auf IT-Systeme. Sie wissen um die verfügbaren Werkzeuge und Methoden im Bereich der offensiven Sicherheit. Sie können einen offensiven Sicherheitstest methodisch durchführen und die Sicherheit eines Zielsystems adäquat bewerten. Sie erkennen verschiedene Schwachstellentypen in Web-Applikationen und wie diese ausgenutzt werden könnten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die wichtigsten Angriffsmethoden gegen IT-Systeme
- die gängige Angriffswerkzeuge
- typische Schwachstellen und Verteidigungsmaßnahmen in IT-Systemen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- Angriffswerkzeuge effektiv und sicher einzusetzen
- das Risiko eines Angriffs abzuwägen und ggf. auf Angriffe zu verzichten
- Schwachstellen zu finden und zu bewerten
- geeignete Gegenmaßnahmen vorzuschlagen
- einen Penetrationstest-Bericht zu erstellen

Übergreifende Kompetenzen

Die Studierenden können

- offensive Sicherheitsüberprüfungen durchführen und die Sicherheit von IT-Systemen bewerten

Inhalt:

- Pentest Methodik und vertiefte rechtliche Grundlagen
- Typische Schwachstellen in IT-Systemen
- Angriffstypen, Angriffsvektoren, Top 10 der gängigen Web-Angriffe
- Angriffe gegen Windows und Linux Systeme

- Die wichtigsten Werkzeuge des Penetration Testing
- Praktische Durchführung von Angriffen

Literaturhinweise:

- Messner, Michael: Hacking mit Metasploit: Das umfassende Handbuch zu Penetration Testing und Metasploit, dpunkt.verlag GmbH, 3. Auflage, 2017.
- Kim, Peter: The Hacker Playbook: Practical Guide to Penetration Testing,. CreateSpace Independent Publishing Platform, 2014.
- Kim, Peter: The Hacker Playbook: Practical Guide to Penetration Testing, CreateSpace Independent Publishing Platform, 2015.
- Kim, Peter: The Hacker Playbook3: Practical Guide to Penetration Testing, CreateSpace Independent Publishing Platform, 2018.
- Stuttard, Dafydd, Pinto, Marcus. Wiley, John & Sons: The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, , 2. Auflage, 2011 (ISBN 978-1118026472).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können offensive Werkzeuge und deren Wirkungsweise richtig einschätzen und eigenständig anwenden. Darüber hinaus sind sie befähigt, IT-Systeme zu analysieren und mittels eines selbst angefertigten Penetrationstest-Berichts zu bewerten.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Projekt IT-Sicherheit

Schlüsselwörter: Teamprojekt, Projektmanagement, Softwareentwicklung, Sicherheitsanalyse, Sicherheitstest

Zielgruppe:	**4. Semester ISB**	**Modulnummer:**	**IT 105 XXXX**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**30 h**
	Selbststudium		**120 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Tobias Heer**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

Kenntnisse über Programmiersprachen und Methoden der Softwaretechnik, Grundlagen IT-Sicherheit, Kryptografie

Gesamtziel:

Die Studierenden erwerben die Kompetenz zur Analyse von IT-Systemen und Software bzw. zur Entwicklung sicherer Software-Anwendungen und Sicherheitsanwendungen.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz zur Analyse und Entwicklung komplexer Software- Anwendungen mit Sicherheitsbezug. Sie können das bereits erworbene Wissen im Kontext eines Projekts im Rahmen eines umfangreichen Tests oder einer Software-Entwicklungsaufgabe anwenden und vertiefen. Sie beherrschen die methodische Vorgehensweise der Sicherheitsanalyse und Software-Entwicklung. Des Weiteren sind sie in der Lage, Methoden und Techniken aus dem Bereich Soft Skills anzuwenden.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Methoden der Sicherheitsüberprüfung eines IT-Systems
- die Methoden der sicheren Software-Entwicklung

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- eine komplexe Software oder ein komplexes System in hinsichtlich seiner verwendeten Software, seiner Schnittstellen und seiner Sicherheitsanforderungen analysieren
- die im Studium erlernten Methoden der Sicherheitsanalyse und des Sicherheitstests einsetzen
- die gewonnenen Erkenntnisse strukturiert in einem Bericht darstellen
- das erarbeitet Ergebnis vor einer Gruppe sicher präsentieren

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- methodische Vorgehensweisen der professionellen Sicherheitsanalyse und sicherheitsbezogenen Software-Entwicklung einzusetzen

Inhalt:

- Projektmanagement und Teamarbeit
- Arbeitstechniken des Zeitmanagements, der Arbeitsorganisation und Informationsgewinnung/-recherche

- Wissenschaftliches Arbeiten
- Kommunikation und Präsentation
- Technische Dokumentation
- Softwaretechnik:
  Security-Requirements, Gefahrenmodellierung, Anforderungsanalyse, Design, Implementierung, Test, Installation

Literaturhinweise:

- Richter, Ludewig: Software Engineering. dpunkt Verlag, 2013.
- Yaworski, Peter: Hacking und Bug Hunting: Wie man Softwarefehler aufspürt und damit Geld verdient - ein Blick über die Schulter eines erfolgreichen Bug Hunters, dpunkt.verlag GmbH , 2020.
- Frank, Simon: Basiswissen Sicherheitstests: Aus- und Weiterbildung zum ISTQB Advanced Level Specialist - Certified Security Tester, dpunkt.verlag, 2019.

Wird angeboten:

in jedem Semester

Lehr- und Lernform:	Teamprojekt
Leistungskontrolle:	Bericht (schriftlich) und Referat (20 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lernziele:

Die Studierenden können methodische Vorgehensweisen der professionellen Software-Entwicklung einsetzen. Weiterhinkönnen sie Software und IT-Systeme bezüglich Ihrer Sicherheit bewerten.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernziele:

Die Studierenden beherrschen die konzeptionelle Aufteilung eines komplexen IT-Systems zum besseren Verständnis und als Vorbereitung zum Entwurf oder der Analyse. Die Studierenden beherrschen Vorgehensweisen zur Verbesserung der persönlichen Fertigkeiten. Die Studierenden verfügen über Kenntnisse zur Rollenverteilung im Projekt-Team und dessen Gruppendynamik. Sie können das erarbeitet Ergebnis vor einer Gruppe sicher präsentieren.

Bildung der Modulnote:

Bericht und Referat benotet, unbenotetes Testat

Modulbeschreibung Softwarearchitektur

Schlüsselworte: Architekturen, Objektorientierte Modellierung

Zielgruppe:	**4. Semester SWB, ISB**	**Modulnummer:**	**IT 105 4007**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr. Jörg Friedrich**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

- Objektorientierte Systeme
- UML 2

Modulziel – angestrebte Lernergebnisse:

Die Studierenden können Anforderungen, auch abgeleitete Anforderungen in komplexe Softwarearchitekturen umsetzen. Sie setzen passende Entwurfs- und Architekturmuster, sowie Frameworks und Bibliotheken ein. Sie besitzen die Kompetenz für ein ingenieurmäßiges Vorgehen bei der Erstellung der Software-Applikation.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Frameworks und Bibliotheken für SOA
- Entwurfsmuster

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Entwurfsmuster auswählen und anwenden
- Webservices programmieren

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- Probleme im Bereich Softwarearchitektur zu lösen, sowie die Auswahl von Software-Technologien zu bewerten

Inhalt:

- Architektur und Architekten
- Vorgehen bei der Architekturentwicklung
- Architekturschichten, UML 2 für Architekten
- Objektorientierte Entwurfsprinzipien
- Architektur- und Entwurfsmuster
- Technische Aspekte, Berücksichtigung von Anforderungen und Randbedingungen
- Middleware, Frameworks, Referenzarchitekturen, Modell-getriebene Architektur
- Komponenten, Komponententechnologien, Schnittstellen (API)
- Bewertung von Architekturen
- Refactoring, Reverse Engineering

Literaturhinweise:

- Goll, J.: Methoden der Softwaretechnik, Vieweg-Teubner, 2012.
- Goll, J. & Dausmann, M.: Architektur- und Entwurfsmuster, Vieweg-Teubner, tbp 2013.
- Starke, G.: Effektive Softwarearchitekturen, Hanser, 2011.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können Entwurfs- und Architekturmuster auswählen und anwenden. Sie sind in der Lage, Komponenten (EJB) sowie Webservices (SOA) zu programmieren.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Praktisches Studiensemester

Schlüsselwörter: Praktische Ingenieurerfahrung im industriellen Umfeld, Projektarbeit im Team

Zielgruppe:	**5. Semester WKB, SWB, ISB**	**Modulnummer:**	**IT 105 5000**
Arbeitsaufwand:	**26 ECTS**		**780 h**
Davon	**Kontaktzeit**		**780 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Kai Warendorf**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Abgeschlossener erster Studienabschnitt

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden zum ingenieursmäßigen Arbeiten auf dem Gebiet der Softwaretechnik befähigt. Die Studierenden beherrschen das ingenieurmäßige Arbeiten in einem Projektteam.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- den organisatorischen Aufbau und Funktionsweise einer Abteilung

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- die Methoden des Projektmanagement anwenden
- die im Studium erlernten Modelle und Methoden zur Lösung berufspraktischer Problemstellungen anwenden

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- sich im industriellen Umfeld einer Firma sicher zu bewegen
- Lösungswege der Praxis auf Basis der im Studium entwickelten Kompetenzen kritisch zu reflektieren

Inhalt:

100 Tage betriebliche Praxis in einem Unternehmen oder einer Firma aus dem IT-Bereich

Literaturhinweise:

- Hering, Lutz, Hering, Heike, Heyne, Klaus-Geert: Technische Berichte, Vieweg, 2014.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Praktikum
Leistungskontrolle:	Bericht, Referat (20 Minuten)
Anteil Semesterwochenstunden:	26 SWS
Geschätzte studentische Arbeitszeit:	780 Stunden

Lernziele:

Die Studierenden beherrschen das ingenieurmäßige Arbeiten in einem Projektteam.

Bildung der Modulnote:

unbenotetes Testat

Modulbeschreibung Schlüsselqualifikationen

Schlüsselworte: **Berufsstart, Wissenschaftliches Arbeiten, Disputation, Technisches Englisch**

Zielgruppe:	**5. Semester WKB, SWB, ISB**	**Modulnummer:**	**IT 105 5001**
Arbeitsaufwand:	**4 ECTS**		**120 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Andreas Rößler**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Schulkenntnisse in Englisch

Modulziel – angestrebte Lernergebnisse:

Studierenden erwerben Kompetenzen in

- Kommunikationsfähigkeit,
- Disputation,
- Fremdsprachen,
- wissenschaftlichen Schreiben,
- Bewerbungsverfahren.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- adäquates und situationsbezogenes berufliches Handeln.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- den Berufsstart erfolgreichen durchzuführen
- sich sicher im beruflichen Umfeld bewegen
- wissenschaftliche Artikel erstellen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- wissenschaftliche Texte über ingenieurwissenschaftlich Themen auch in englischer Sprache zu erstellen
- auch in englischer Sprache sicher zu kommunizieren

Inhalt:

Wissenschaftliches Arbeiten

- Strukturieren
- Recherchieren
- Analysieren
- Wissenschaftliche Schreiben und Zitieren

Berufsstart

- Karriereplanung
- Bewerbertraining

Technisches Englisch

- TOEFL-Test

Literaturhinweise:

- Stemmer, B., Wynne, T.: Grammar Rules. Grundlagen der englischen Grammatik, Klett Verlag, 2000.
- Schulz von Thun, F.: Miteinander reden, Band 1-3, Rowohlt TB, 2008.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung und Übungen
Leistungskontrolle:	Hausarbeit und Referat (20 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernziele:

Die Studierenden erwerben und vertiefen die Fähigkeit zur inhaltlichen Erfassung und Erstellung ingenieurwissenschaftlicher Texte.

Lehr- und Lernform:	TOEFL-Vorbereitungskurs
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden erwerben die Fähigkeit zur inhaltlichen Erfassung technisch-wissenschaftlicher Texte und zur Kommunikation über technisch- wissenschaftliche Themen in englischer Sprache.

Bildung der Modulnote:

unbenotetes Testat

Modulbeschreibung Cyber-Physical Networks

Schlüsselworte: Industrielle Netzwerke, Fahrzeugvernetzung, Time-Sensitive Networking, Edge-Computing

Zielgruppe:	**6. Semester TIB-CPS, ISB**	**Modulnummer:**	**TIB 105 6035**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**75 h**
	Selbststudium		**45 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Englisch**		
Modulverantwortung:	**Prof. Dr.-Ing. Michael Scharf**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

- Grundlegende Kenntnisse zu Rechnernetzen
- Grundlegende Kenntnisse zu Betriebssystemen
- Gute Kenntnisse Software-Engineering

Modulziel – angestrebte Lernergebnisse:

Die Studierenden können die Vernetzung von Cyber-Physischen Systemen verstehen. Sie beherrschen die verschiedenen Aspekte der Vernetzung von Cyber-Physischen Systemen. Sie sind in der Lage diese zu konzipieren und zu betreiben.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Anforderungen und Lösungen zur Echtzeit-Kommunikation,
- wesentliche Systeme für industrielle Netzwerke und Fahrzeugvernetzung,
- relevante Standards wie Time-Sensitive Networking (TSN) bzw. OPC UA,
- Internet-of-Things (IoT) Lösungen zur internetbasierten Vernetzung.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Cyber-Physische Netzwerke zu verstehen und zu bewerten,
- Netzwerke zu konfigurieren und Komponenten in ein System einzubinden.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Cyber-Physische Systeme gesamthaft verstehen, bewerten und beherrschen.

Inhalt:

- Anforderungen und Konzepte der Echtzeit-Kommunikation
- Architekturen für industriellen Netzwerke und Fahrzeugvernetzung
- Quality-of-Service (QoS) Mechanismen für Echtzeit-Kommunikation
- Beispiele zu bedeutenden Systemen und Protokollen wie CAN, Industrial und Automotive Ethernet, Time-Sensitive Networking (TSN), OPC UA.
- Funknetzwerke für das Internet of Things (IoT)
- Netzwerk-Planung, Betrieb und Optimierung, Edge-Computing
- Technologien und Standards für das Netzwerkmanagement

Literaturhinweise:

- Tanenbaum, A.; Feamster, N.; Wetherall, D.: Computer Networks, 6th Edition, Pearson, 2021
- Klasen, F.(Hrsg.): Industrielle Kommunikation mit Feldbus und Ethernet. VDE-Verlag , 2010
- Zimmermann, W.; Schmidgall, R.: Bussysteme in der Fahrzeugtechnik: Protokolle, Standards und Softwarearchitektur. Vieweg + Teubner, 5. Auflage, 2014.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können vernetzte Cyber-Physische Systeme konzipieren und betreiben.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Informationssysteme

Schlüsselworte: Informationssysteme

Zielgruppe:	**6. Semester WKB, ISB**	**Modulnummer:**	**IT 105 6001**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Dirk Hesse**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Kenntnisse in

- Programmieren
- Objektorientierte Systeme
- Softwaretechnik
- Datenbanken

Modulziel – angestrebte Lernergebnisse:

Vermittelt wird die Fähigkeit, Informationssysteme mit relationaler Datenhaltung von der Problemanalyse und Anforderungsdefinition über den Architekturentwurf bis hin zur Programmentwicklung und dessen Test zu entwerfen, zu entwickeln und zu betreiben.
Die Studierenden verstehen die grundlegenden Konzepte des Entwurfs, der Entwicklung und des Betriebs von Informationssystemen. Sie können die Methoden und Vorgehensweisen der Informationssystemgestaltung mit den zugehörigen Schichtenarchitekturen und Datenmodellen in der Praxis anwenden. Sie sind in der Lage, lauffähige Anwendungen mit Hilfe von Entwicklungsplattformen und CASE generierten Datenbankmodellen zu erstellen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Architekturen integrierter Informationssysteme
- UML, ERM, Datenmodelle, Normalformtheorien
- Architektur- und Datenbankmodelle im Rahmen des Architekturentwurfs
- Entwicklungswerkzeuge, Sprachen und Bibliotheken, Entwicklungsplattformen
- Datenbanksysteme und Datendienste

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- Problemanalysen und Anforderungsdefinitionen durchzuführen
- Architektur- und Datenbankmodelle mit Hilfe von CASE Tools zu entwerfen
- Programmentwicklungen mit Hilfe von Entwicklungsplattformen durchzuführen
- Datenbanksysteme und Cloudanbindungen zu nutzen

Übergreifende Kompetenzen

Die Studierenden können

- Informationssysteme planen und entwickeln
- die Digitale Transformation durch Anwendung digitaler Technologien umsetzen

Inhalt:

- Problemanalyse und Anforderungsdefinition
- Architekturentwurf: Architekturmodelle und Schichtenmodelle
- Normalformtheorien
- Programmentwicklung und Test: Entwicklungswerkzeuge, Entwicklungsplattformen
- Nutzung von Datenbanksystemen und Datendiensten

Literaturhinweise:

- Wallace, P.: Introduction to Information Systems: People, Technology and Processes (3rd Edition), 2019.
- Stair, Reynolds: Principles of Information Systems (Englisch), 13. Auflage, 2019.
- Connolly, T.: Database Systems: A Practical Approach to Design, Implementation and Management, 2014.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Workshop
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernergebnisse:

Die Studierenden beherrschen Analyse, Design und Implementierung einer Anwendung zur Ressourcenplanung.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Security Management und Datenschutz

Schlüsselworte: Informationssicherheitsmanagementsystem (ISMS), Datenschutzrecht, Anonymität, Privacy Enhancing Technologies

Zielgruppe:	**6. Semester ISB**	**Modulnummer:**	**ISB xxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch / Englisch**		
Modulverantwortung:	**Prof. Dr. Dominik Schoop**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

Kenntnisse in Netzwerken, IT-Sicherheit undKryptographie

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, die Compliance eines sozio-technisches System bzgl. IT-Sicherheit und Datenschutz zu bewerten, um möglichen Handlungsbedarf abzuleiten. Die Studierenden können für ein IT-System einschließlich seiner Nutzer und Nutzungsprozesse notwendige Vorgaben machen und notwendige Prozesse aufsetzen, um das IT-System sicherer zu machen und datenschutzgerecht zu gestalten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- den PDCA-Zyklus
- die Komponenten eines Informationssicherheitsmanagementsystems (ISMSs)
- die Anforderungen aus den Standards für ein ISMS
- die Grundzüge des europäischen und deutschen Datenschutzrechts
- datenschutzrechtliche Anforderungen an ein sozio-technisches System
- Möglichkeiten der datenschutzfreundlichen Technikgestaltung (z.B. Anonymisierung)
- Techniken der Datensicherheit (Zutritts-, Zugangs-, Zugriffsschutz, Kryptographie)
- den Bezug eines ISMS zu einem Datenschutzmanagementsystem (DSMS)

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage,

- ein sozio-technisches System bzgl. seiner Bedürfnisse für ein ISMS zu analysieren
- die Themen eines ISMS Komponenten eines IT-Systems zuzuordnen
- ein ISMS nach ISO 27001ff bzw. nach BSI-Grundschutz zu entwerfen
- ausgewählte Teile eines ISMS zu implementieren
- personenbezogene Daten zu erkennen
- die geeignete Rechtsgrundlage auszuwählen
- ein Verzeichnis der Verarbeitungstätigkeit zu erstellen
- eine Datenschutzfolgenabschätzung durchzuführen
- Datensätze angemessen zu anonymisieren

Übergreifende Kompetenzen

Die Studierenden können

- nach Analyse eines sozio-technischen IT-Systems nachhaltige Wege aufzeigen, wie das IT-System den Anforderungen der IT-Sicherheit und des Datenschutzrechts genügen kann.

Inhalt:

- Grundbegriffe (Assets, Schutzbedarf, Bedrohung, Risiko, personenbezogene Daten, Privacy, Anonymität, Pseudonymität, Unlinkability, Unobservability)

- PDCA-Zyklus für Informationssicherheit
- Systemmodellierung (Strukturanalyse, Schutzbedarf, Bedrohungsanalyse)
- Ansätze zur Risikobewertung
- Informationssicherheit nach ISO 27001ff bzw. nach BSI-Grundschutz
- datenschutzrechtliche Grundlagen (DSGVO, LDSG, BDSG, EU-Privacy-Verordnung)
- Anonymität & Pseudonymität (lokale und globale Anonymität, k-Anonymität, Differential Privacy, Anonymität im Netz)
- Kryptographische Verfahren für den Datenschutz
- Privacy Enhancing Technologies
- Beispiele zu Privacy by Design (anonyme Zahlungen, elektronische Wahlen, elektronischer Reisepass, Web-Tracking, Messaging-Systeme)

Literaturhinweise:

- DIN EN ISO/IEC 2700x IT-Sicherheitsverfahren – Informationssicherheits-Managementsysteme
- Bundesamt für Sicherheit in der Informationstechnik (BSI), BSI-Standard-200-1/2/3 „BSI-Grundschutz“
- Voigt, P. & von dem Bussche, A.: EU-Datenschutz-Grundverordnung (DSGVO): Praktikerhandbuch, Springer, 2018.
- Petrlic, R. & Sorge, Chr.: Datenschutz: Einführung in technischen Datenschutz, Datenschutzrecht und angewandte Kryptographie, Springer-Vieweg, 2017.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Secure Software Development

Schlüsselworte: Software, Secure Software Development Lifecycle

Zielgruppe:	**6. Semester ISB**	**Modulnummer:**	**ISB xxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch / Englisch**		
Modulverantwortung:	**Prof. Dr. Dominik Schoop**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

Computerarchitektur, Betriebssysteme, Programmieren, Softwaretechnik, Softwarearchitektur, Software Testing, IT-Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden sind befähigt, mit einem zielgerichteten Entwicklungsprozess sichere Software zu entwickeln.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- Software-Entwicklungsmethoden (klassisch, agil)
- die Phasen und Komponenten von Development Operations (DevOps)
- Softwarearchitekturen und Entwurfs- und Architekturmuster
- Arten und Sicherheitsfolgen von Softwarefehlern (Designfehler, Implementierungsfehler)
- Software Security Design Patterns
- sichere Programmierung
- Testmethoden für Sicherheitsfehler (statische und dynamische Analyse)

Fertigkeiten – methodische Kompetenzen
Die Studierenden sind in der Lage,

- Methoden und Schritte für die Phasen des DevOps zu definieren, die Softwaresysteme sicherer machen
- Sicherheitsanforderungen an ein Softwaresystem zu definieren
- Design- und Implementierungsfehler in Source Code ausgewählter Programmiersprachen zu identifizieren und zu korrigieren
- geeignete Security Design Patterns für Software auszuwählen und in ausgewählten Programmiersprachen zu implementieren
- Sicherheitstests von Software zu definieren und praktisch durchzuführen
- Software-Entwicklungswerkzeuge für die Entwicklung sicherer Software einzusetzen

Übergreifende Kompetenzen
Die Studierenden können:

- sowohl die Prozesse als auch die technischen Details der Softwareentwicklung so gestalten, dass sichere Software entwickelt wird und betrieben werden kann.

Inhalt:

- Vorgehensweisen in der Softwareentwicklung
- Development Operations
- Softwarearchitektur
- Anforderungen an sichere Software und den Entwicklungsprozess
- Sicherheitstechniken (Authentisierung, Autorisierung, Accounting, Rechtekonzepte, Separation)
- Entwurfs- und Architekturmuster für sichere Systeme

- Umsetzung von Software Security Design Patterns
- Erkennen und Vermeiden von Softwarefehlern in ausgewählten Programmiersprachen
- statische und dynamische Sicherheitstests von Software

Literaturhinweise:

- Anderson, R.: Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, 2021.
- Faily, S.: Designing Usable and Secure Software with IRIS and CAIRIS, Springer, 2018.
- Fernandez-Buglioni, E.: Security Patterns in Practice: Designing Secure Architectures Using Software Patterns, Wiley, 2013.
- Deogun, D. et al.: Secure by Design, Manning, 2019.
- Schumacher, M. et al.; Security Patterns: Integrating Security and Systems Engineering, Wiley, 2005.
- Wilson, G.: DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement, Rethink Press, 2020.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lehr- und Lernform:	Labor
Leistungskontrolle:	Bericht
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Software Testing

Schlüsselworte: Testen, Qualitätssicherung

Zielgruppe:	**6. Semester SWB-SWT, ISB**	**Modulnummer:**	**IT 105 NNNN**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**120 h**
	Selbststudium		**30 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr. Mirko Sonntag**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Prinzipien des Software-Engineering und Kenntnisse in einer objektorientierten Programmiersprache.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, durch qualitätssichernde Maßnahmen die Erfüllung der funktionalen und nicht-funktionalen Anforderungen an Software zu gewährleisten. Sie beherrschen Software-Tests als wichtigstes Mittel der Qualitätssicherung. Sie können Kosten, Nutzen und Grenzen von Software-Tests bei der Entwicklung von Test-Konzepten berücksichtigen und selbstständig Tests entwickeln.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- Organisation und Management von Software-Tests
- Verschiedene Ebenen und Methoden von Software-Tests
- Nutzen und Kosten der Automatisierung von Tests
- Grenzen der Testautomatisierung

Fertigkeiten – methodische Kompetenzen
Die Studierenden sind in der Lage,

- Software-Tests auf verschiedenen Ebenen zu formulieren
- Unit Tests und Integrationstests zu entwickeln und Testobjekte zu isolieren
- Automatisierte Tests im Entwicklungsablauf einzubinden
- Systemtests toolgestützt durchzuführen

Übergreifende Kompetenzen
Die Studierenden können:

- mithilfe von Tests sicherstellen, dass Software von hoher Qualität entwickelt wird

Inhalt:

- Motivation für Qualitätssicherung und Testen
- Testautomatisierung, Testdokumentation und Testmanagement
- Testwerkzeuge
- Black Box und White Box Testing
- Unit Tests und zugehörige Methodiken, wie Mocking und Test-driven Development
- Integrationstests
- System Tests (Performance & Load Testing, Penetration Tests)
- Akzeptanztests
- GUI Tests

Literaturhinweise:

- Baumgartner et al.: Agile Testing, Hanser, 2. Auflage, 2018.
- Crispin, Lisa & Gregory, Janet: Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley, 2008.
- Kinsbruner, Eran: The Digital Quality Handbook: Guide for Achieving Continuous Quality in a DevOps Reality, Infinity P, 2017.
- Rasmusson, Jonathan: The Way of the Web Tester, O'Reilly, 2016.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernziele:

Die Studierenden sind in der Lage, automatische Tests auf verschiedenen Ebenen der Testpyramide zu implementieren und in die Continuous Integration Pipeline zu integrieren. Darüber hinaus, können Sie toolgestützte Systemtests durchführen, um die Einhaltung nicht-funktionaler Anforderungen von Software sicherzustellen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Digitale Forensik

Schlüsselworte: Forensik, digitale Spuren, Beweissicherung

Zielgruppe:	**6. Semester ISB**	**Modulnummer:**	**ISB xxx**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**60 h**
	Selbststudium		**60 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch / Englisch**		
Modulverantwortung:	**Prof. Dr. Dominik Schoop**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.07.2022**		

Empfohlene Voraussetzungen:

Computerarchitektur, Betriebssysteme, IT-Sicherheit

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, mit wissenschaftlichen Methoden digitale Spuren in IT-Systemen gerichtsfest zu analysieren und zu sichern.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die Rechtslage für den Zweck und den Einsatz digitaler Forensik
- rechtliche und technische Anforderungen an verwertbare digitale Spuren
- die Funktionsweise von ausgewählten Dateisystemen und wie Aktivitäten dort dokumentiert werden
- die Art und Weise wie und wo Betriebssysteme und ausgewählte Applikationen benutzerbezogene Daten speichern

Fertigkeiten – methodische Kompetenzen
Die Studierenden sind in der Lage,

- die digitale Forensik in den rechtlichen Rahmen einzuordnen
- digitale Spuren aus Dateisystemen zu extrahieren
- gelöschte Dateien wiederherzustellen
- digitale Spuren aus Geräten mit gängigen Betriebssystemen und ausgewählten Applikationen zu extrahieren

Übergreifende Kompetenzen
Die Studierenden können:

- digitale Spuren von IT-Systemen sichern

Inhalt:

- Motivation und Begriffsbestimmungen
- digitale Spuren
- forensisches Vorgehen (Datensicherung, Datenaufbereitung, Rekonstruktion, Reduktion, Analyse)
- Mechanismen persistenter Speicher
- Funktionsweise von Dateisystemen (FAT, NTFS, EXT)
- Analyse von Dateisystemen, Wiederherstellung von Daten
- Analyse von Spuren in Betriebssystemen (Windows, Linux, Android und iOS)
- Spuren in ausgewählten Applikationen (z.B. Messenger und Browser)
- Analyse flüchtiger Spuren (Netzwerk)

Literaturhinweise:

- Altheide, C. & Carvey, H.: Digital Forensics with Open Source Tools, Syngress, 2011.
- Dewald, A. & Freiling, F.C.: Forensische Informatik, BoD, 2015.
- Kuhlee, L & Völzow, V.: Computer-Forensik Hacks, O'Reilly, 2012.

- Labudde, D. & Spranger, M.: Forensik in der digitalen Welt, Springer, 2017.
- Vacca, J.: Computer Forensics: Computer Crime Scene Investigation, Jones & Bartlett Publ., 2022.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lehr- und Lernform:	Labor
Leistungskontrolle:	Bericht
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Studienprojekt

Schlüsselworte: Softwaretechnik

Zielgruppe:	**6. Semester SWB, ISB**	**Modulnummer:**	**IT 105 6007**
Arbeitsaufwand:	**5 ECTS**		**150 h**
Davon	**Kontaktzeit**		**5 h**
	Selbststudium		**135 h**
	Prüfungsvorbereitung		**10 h**
Unterrichtssprache:	**Deutsch**		
Modulverantwortung:	**Prof. Dr.-Ing. Reinhard Schmidt**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Abgeschlossener erster Studienabschnitt

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, ein ingenieurwissenschaftliches Projekt auf dem Gebiet der Softwaretechnik zu bearbeiten.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die im Studium erlernten Modelle und Methoden zur Lösung ingenieurwissenschaftliche Problemstellungen

Fertigkeiten – methodische Kompetenzen
Die Studierenden können:

- Zeit- und Projektmanagement
- wissenschaftliches Arbeiten und Schreiben
- wissenschaftliches Präsentieren

Übergreifende Kompetenzen
Die Studierenden sind in der Lage,

- selbstständig wissenschaftlich zu arbeiten

Inhalt:

Im Studienprojekt ist unter Anleitung eines/einer betreuenden ProfessorIn, eine ingenieurmäßige Aufgabenstellung aus dem Gebiet der Softwaretechnik zu lösen.

Literaturhinweise:

- Hering, Lutz & Hering, Heike: Technische Berichte, Vieweg, 2017 (ISBN 978-3-8348-15-86-6).
- Heesen, Bernd: Wissenschaftliches Arbeiten, Springer Verlag, 2014 (ISBN 978-3-662-43346-1),
- Lobin, Henning: Die wissenschaftliche Präsentation: Konzept – Visualisierung – Durchführung; Schönigh Verlag, 2012 (ISBN 978-3-3770-7).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	5 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage, eine Problemstellung selbstständig wissenschaftlich zu bearbeiten.

Bildung der Modulnote:

benoteter Bericht und Referat

Modulbeschreibung Bachelorarbeit

Schlüsselwörter: **Abschlussarbeit, wissenschaftliches und ingenieurmäßiges Arbeiten, Projektarbeit**

Zielgruppe:	**7. Semester WKB, ISB**	**Modulnummer:**	**IT 105 7000**
Arbeitsaufwand:	**15 ECTS**		**450 h**
Davon	**Kontaktzeit**		**40 h**
	Selbststudium		**340 h**
	Prüfungsvorbereitung		**70 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr.-Ing. Reinhard Schmidt**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

- alle Prüfungsleistungen der ersten vier Semester müssen erfolgreich abgeschlossen sein
- abgeschlossenes Praxissemester
- fundierte Kenntnisse im eigenen Studienprofil

Modulziel – angestrebte Lernergebnisse:

Die Studierenden besitzen die Fähigkeit, sich in ingenieurmäßige Fragestellungen aus dem Bereich der Informatik einzuarbeiten. Sie können wissenschaftliche und technische Weiterentwicklungen verstehen und auf Dauer verfolgen.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die Vorgehensweisen beim wissenschaftlichen Arbeiten

Fertigkeiten – methodische Kompetenzen
Die Studierenden können:

- systematische Recherchen zu ingenieurwissenschaftlichen Fragestellungen durchführen

Übergreifende Kompetenzen
Die Studierenden erlangen

- detaillierte Einblicke und umfassende Erkenntnisse auf einem Teilgebiet der Technischen Informatik. Softwaretechnik oder IT-Sicherheit

Inhalt:

In der Bachelorarbeit soll den Studierenden zeigen, dass die während des Studiums erlernten Kenntnisse und erworbenen Fähigkeiten erfolgreich in die Praxis umgesetzt werden können. Dazu wird eine projektartige Aufgabe unter Einsatz von ingenieurmäßigen Methoden bearbeitet. Der/die betreuende ProfessorIn begleitet die Studierenden während der Bachelorarbeit und leitet sie zum wissenschaftlichen Arbeiten an. Die Arbeit schließt mit einer schriftlichen Ausarbeitung und einem Vortrag ab.

Literaturhinweise:

- Hering, Lutz & Hering, Heike: Technische Berichte, Vieweg, 2017 (ISBN 978-3-8348-15-86-6).
- Heesen, Bernd: Wissenschaftliches Arbeiten, Springer Verlag, 2014 (ISBN 978-3-662-43346-1),
- Lobin, Henning: Die wissenschaftliche Präsentation: Konzept – Visualisierung – Durchführung; Schönigh Verlag, 2012 (ISBN 978-3-3770-7).Brink, Alfred: Anfertigung wissenschaftlicher Arbeiten, Springer Gabler Verlag, 2013 (ISBN 978-3-8349-4396-5).
- Müller, Ragnar, Plieninger, Jürgen, Rapp, Christian: Recherche 2.0, Springer Verlag, 2013 (ISBN 978-3- 658- 02249-5).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Selbstständiges wissenschaftliches Arbeiten
Leistungskontrolle:	Bericht
Anteil Semesterwochenstunden:	12 SWS
Geschätzte studentische Arbeitszeit:	360 Stunden

Lernziele:

Die Studierenden beherrschen selbstständiges wissenschaftliches Arbeiten. Sie erwerben die Fähigkeit zum wissenschaftlichen und ingenieurmäßigen Arbeiten, sowohl eigenständig als auch im Projekt-Team.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Präsentation einer wissenschaftlichen Arbeit
Leistungskontrolle:	Referat (20 Minuten) Testat, Teilnahme am IT-Kolloquium
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernziele:

Die Studierenden können ihre eigene wissenschaftliche Arbeit präsentieren und überzeugend argumentieren.

Bildung der Modulnote:

gemittelte Note aus Bericht, Faktor 12 und Referat Faktor 3
unbenotetes Testat

Modulbeschreibung Wahlfachmodul

Schlüsselwörter: Vertiefung im eigenen Studienprofil

Zielgruppe:	**7. Semester WKB, ISB**	**Modulnummer:**	**MD 7630**
Arbeitsaufwand:	**6 ECTS**		**180 h**
Davon	**Kontaktzeit**		**120 h**
	Selbststudium		**30 h**
	Prüfungsvorbereitung		**30 h**
Unterrichtssprache:	**Deutsch oder Englisch**		
Modulverantwortung:	**Prof. Dr.-Ing. Reinhard Schmidt**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Grundlegende Kenntnisse im eigenen Studienprofil

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erlangen eine wissenschaftliche und fachliche Vertiefung auf dem Gebiet der Softwaretechnik.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- aktuelle und industrienahe Techniken

Fertigkeiten – methodische Kompetenzen
Die Studierenden können:

- aktuelle und industrienahe Techniken anwenden

Übergreifende Kompetenzen
Die Studierenden sind in der Lage,

- aktuelle und industrienahe Techniken zu implementieren

Inhalt:

Das Wahlfachmodul besteht aus Wahlpflichtfächern mit einem Umfang von insgesamt 6 SWS. Studierende wählen zur Vertiefung seines Studienprofils drei Wahlfächer mit jeweils 2 SWS. Die zur Auswahl stehenden Wahlpflichtfächer werden zu Semesterbeginn öffentlich bekannt gegeben.
In den Wahlpflichtfächer werden aktuelle und industrienahe Techniken angeboten.

Literaturhinweise:

abhängig vom gewählten Wahlpflichtfach

Wird angeboten:

Wahlpflichtfächer werden jährlich angeboten.
Alle Wahlpflichtfächer sind im Modulhandbuch der Wahlpflichtfächer beschrieben.
Der Angebotsrhythmus ist ebenfalls im Modulhandbuch der Wahlpflichtfächer festgelegt.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	abhängig vom gewählten Wahlpflichtfach
Leistungskontrolle:	abhängig vom gewählten Wahlpflichtfach
Anteil Semesterwochenstunden:	3 x 2 SWS
Geschätzte studentische Arbeitszeit:	180 Stunden

Lernziele:

Die Studierenden verfügen über eine wissenschaftliche und fachliche Vertiefung im eigenen Studienprofil.

Bildung der Modulnote:

Mittelwert der Noten der Wahlpflichtfächer

Modulbeschreibung Wissenschaftliche Vertiefung

Schlüsselwörter: Eigenständiges Arbeiten in Entwicklung und Forschung

Zielgruppe:	**7. Semester WKB, ISB**	**Modulnummer:**	**IT 105 7001**
Arbeitsaufwand:	**9 ECTS**		**270 h**
Davon	**Kontaktzeit**		**20 h**
	Selbststudium		**210 h**
	Prüfungsvorbereitung		**40 h**
Unterrichtssprache:	**Deutsch / Englisch**		
Modulverantwortung:	**Prof. Dr.-Ing. Reinhard Schmidt**		
Dauer des Moduls:	**ein Semester**		
Stand:	**01.09.2019**		

Empfohlene Voraussetzungen:

Fundierte Kenntnisse im eigenen Studienprofil

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sich in ingenieurmäßige Fragestellungen aus dem Bereich der Informatik einzuarbeiten, wissenschaftliche und technische Weiterentwicklungen zu verstehen und auf Dauer verfolgen zu können.

Kenntnisse – fachliche Kompetenzen
Die Studierenden kennen:

- die Vorgehensweisen beim wissenschaftlichen Arbeiten

Fertigkeiten – methodische Kompetenzen
Die Studierenden können:

- systematische Recherchen zu ingenieurwissenschaftlichen Fragestellungen durchführen

Übergreifende Kompetenzen
Die Studierenden erlangen

- detaillierte Einblicke und umfassende Erkenntnisse auf einem Teilgebiet der Informatik.

Inhalt:

Recherche und Selbststudium im Umfeld der Bachelorarbeit

Literaturhinweise:

- Heesen, Bernd: Wissenschaftliches Arbeiten, Springer Verlag, 2014 (ISBN 978-3-662-43346-1).
- Brink, Alfred: Anfertigung wissenschaftlicher Arbeiten, Springer Gabler Verlag, 2013 (ISBN 978-3-8349-4396-5).
- Müller, Ragnar, Plieninger, Jürgen, Rapp, Christian: Recherche 2.0, Springer Verlag, 2013 (ISBN 978-3- 658- 02249-5).

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Recherche und Selbststudium
Leistungskontrolle:	Mündliche Prüfung (20 Minuten)
Anteil Semesterwochenstunden:	9 SWS
Geschätzte studentische Arbeitszeit:	270 Stunden

Lernziele:

Die Studierenden können aufgrund eigener Recherchen Problemstellungen der Softwaretechnik analysieren und eigenständig Problemlösungen finden und bewerten.

Bildung der Modulnote:

Mündliche Prüfung