

Universität Paderborn



Informatik  
**Veranstaltungs-  
Kommentar**

Für

Informatik ▷ Bachelor  
▷ Master

Computer Engineering

Ingenieurinformatik

Lehramt Informatik GyGe

*Für das WiSe 17/18*

Von der Fachschaft  
Mathematik/Informatik



# Inhaltsverzeichnis

<b>1</b>	<b>Wichtige Informationen</b>	<b>3</b>
1.1	Benutzerhinweise . . . . .	3
1.2	Literaturangaben . . . . .	3
1.3	Sprechstunden . . . . .	3
1.4	Vollständigkeit . . . . .	3
1.5	Internet . . . . .	3
<b>2</b>	<b>Mitarbeitende der Informatik</b>	<b>4</b>
<b>3</b>	<b>Weitere wichtige Adressen</b>	<b>8</b>
<b>4</b>	<b>Veranstaltungen</b>	<b>9</b>
4.1	Übersicht . . . . .	9
4.2	Informatik . . . . .	13
<b>5</b>	<b>Raum für Notizen</b>	<b>59</b>

# Impressum

Herausgeber: Fachschaft Mathematik/Informatik  
Universität Paderborn, Raum E1.311  
Warburger Straße 100  
33098 Paderborn  
E-Mail: fsmi@uni-paderborn.de  
Telefon: 05251 60-3260

V.i.S.d.P.: Oliver Otte

ISSN: 1868-0690

Redaktion: Oliver Otte, Oliver Butterwegge, Daniela Strotmann, Philipp Breuch, Alex Wiens

Mitarbeitende: die Fachschaft (Korrekturlesen),  
die Dozentinnen und Dozenten der Mathematik und der Informatik (Kommentare)

Auflage: 2<sup>6</sup> Exemplare

# 1 Wichtige Informationen

## 1.1 Benutzerhinweise

zum Kopf:

Name der Veranstaltung
------------------------

**Dozent:** Name des Dozenten

**Büro:** Raum

**Sprechstunde:** Zeit

## 1.2 Literaturangaben

Die Bücher in diesem Abschnitt sind Empfehlungen der Dozenten. Viele dieser Bücher sind in der Bibliothek zu finden, sodass ihr euch die Bücher dort erst ansehen und ausleihen könnt, bevor ihr viel Geld dafür ausgeben müsst. Auf Ebene 3 der Bibliothek befindet sich übrigens der Seminarapparat unserer Fachschaft. In diesem haben wir etwas Grundlagenliteratur zur Informatik und Mathematik gesammelt, welche wir für lesenswert halten.

## 1.3 Sprechstunden

Ein Großteil der Dozentinnen und Dozenten gibt keine feste Sprechstunde mehr an, sondern ist nach Vereinbarung zu sprechen, sowie vor und nach den Veranstaltungen. Daher findet Ihr nicht überall die Angabe einer Sprechstunde.

## 1.4 Vollständigkeit

Da nicht alle Lehrenden einen Veranstaltungskommentar abgegeben haben, ist das Verzeichnis der Veranstaltungen leider nicht vollständig!

## 1.5 Internet

Elektronische Informationen zum Vorlesungsangebot gibt es unter folgenden Adressen:

- <https://cs.uni-paderborn.de/studium/studienangebot/informatik/> – offizielle Webseite zum Studienangebot der Informatik
- <https://math.uni-paderborn.de/studium/studiengaenge/mathematik/> – offizielle Webseite zum Studium der Mathematik
- <http://webptool.cs.upb.de/> – aktuellster Stand der Vorlesungsplanung
- <https://paul.upb.de/> – offizielles Vorlesungsverzeichnis der Uni

Die Seiten der Fachschaft findet Ihr hier: <http://die-fachschaft.de/>

## 2 Mitarbeitende der Informatik

Name	E-Mail	Telefon	Raum
Abshoff, Sebastian	abshoff@upb.de	6433	F1.125
Adelt, Peer	peer.adelt@c-lab.de	6051	FU.234
Affi, Haitham	haitham.affi@upb.de	5373	O3.167
Ahlers, Ulrich	uli@hni.upb.de	6700	F2.320
Allwright, Michael	michael.allwright@uni-paderborn.de	5465/202	ZM1.02-10
Amedick, Justin	justin.amedick@uni-paderborn.de	1727	O2.152
Anis, Anas	anas.anis@upb.de	3309	ZM1.02-11
Anjorin, Anthony	anthony.anjorin@upb.de	5465-224	ZM1.03-06
Antoniou, Paraskewi	antoniou@mail.upb.de	5394	O3.107
Anwer, Jahanzeb	jahanzeb.anwer@upb.de	4342	O3.128
Auroux, Sebastian	sebastian.auroux@upb.de	5385	O3.167
Bauer, Bernard	bb@upb.de	1737	O2.173
Benter, Markus	benter@upb.de	6454	F1.122
Benz, Manuel	manuel.benz@upb.de	3615	ZM1.02-07
Blobel, Johannes	blobel@ccs-labs.org	6494	F U.407
Blömer, Johannes	bloemer@upb.de	6651	F2.101
Blößl, Bastian	bloessl@ccs-labs.org	6494	F U.407
Blume, Bodo	bodo.blume@uni-paderborn.de	1750	O3.173
Bobolz, Jan	jan.bobolz@upb.de	6654	F2.111
Bodden, Eric	eric.bodden@upb.de	3313	ZM1.02-09
Borkowski, Philipp / Azubi	philipp.borkowski@upb.de	1727	O2.152
Boschmann, Alexander	alexander.boschmann@upb.de	5397	O3.119
Böttcher, Stefan	stb@upb.de	6662	F2.217
Brandt, Sascha	myeti@mail.upb.de	6451	F1.203
Brauer, Sascha	sbrauer@upb.de	6653	F2.106
Brunthaler, Stefan	s.brunthaler@upb.de	5380	O3.149
Budde, Lea	lea.budde@upb.de	6613	F2.201
Bunja, Kathrin	kabu@mail.upb.de	6627	F2.106
Bürger, Tanja	tanja.buerger@upb.de	5376	O3.155
Busa-Fekete, Robert,	busarobi@mail.upb.de	3348	O4.170
Buschmeyer, Carmen	carmen@upb.de	6412	F1.426
Buse, Dominik	buse@ccs-labs.org	6560	F1.416
Chvojka, Peter	chvojkap@mail.upb.de		F2.317
Dann, Andreas	andreas.dann@upb.de	5719	ZM1.02-07
Debessay, Seghid	seghid.debessay@c-lab.de	6166	FU.231
Divband Soorati, Mohammad	divband@mail.upb.de	6468	F1.213
Domik, Gitta	domik@upb.de	6610	F2.204
Drächsler, Sevil	sevil.mehraghdam@upb.de	1755	O3.161
Drees, Maximilan	maxdrees@mail.upb.de	6433	F1.125
Dressler, Falko	dressler@ccs-labs.org	6510	F1.401
Eidens, Fabian	feidens@msopb.de	6650	F2.108
Eikel, Benjamin	eikel@upb.de	6452	F1.203
Engels, Gregor	engels@upb.de	3337	ZM1.03.45
Feldkord, Björn	bjoernf@mail.uni-paderborn.de	6434	F1.125

Name	E-Mail	Telefon	Raum
Feldmann, Michael	michael.feldmann@upb.de	6722	F2.403
Feldmann, Rainer	obelix@upb.de	6720	F2.401
Feldotto, Matthias	feldi@mail.upb.de	6452	F1.203
Fischer, Holger	hfischer@s-lab.upb.de	5465/209	ZM1.03-37
Fischer, Matthias	mafi@upb.de	6466	F1.223
Geismann, Johannes	johannes.geismann@upb.de	3309	ZM1.02-11
Gerking, Christopher	christopher.gerking@upb.de	3307	ZM1.02-14
Gerstenberger, Dietrich	dietrich.gerstenberger@upb.de	6611	F2.201
Ghasemzadeh Mohammadi, Hassan	hgm@upb.de	4344	O3.134
Gmyr, Robert	gmyr@mail.upb.de	6704	F2.323
Graf, Tobias	tobias@mail.upb.de	1742	O3.128
Grieger, Marvin	grieger@mail.upb.de	5465/226	ZM1.03-12
Gundelach, Sigrid	sigu@upb.de	6655	F2.207
Hagenauer, Florian	hagenauer@ccs-labs.org	6492	F1.319
Hamann, Heiko	Heiko.Hamann@upb.de	6465	F1.221
Hartel, Rita	rst@upb.de	6612	F2.209
Hasnain, Asif	asif.hasnain@upb.de	1757	O3.164
Hauenschild, Wilfried	wilf@upb.de	5393	O2.146
Haupt, Jutta	jutta@upb.de	3312	ZM1.02.08
Heindorf, Stefan	heindorf@upb.de	5465-207	ZM1.03-07
Heinemann, Birte	birte.heinemann@upb.de	6615	F2.201
Henzgen, Sascha	shenzgen@mail.upb.de	3345	O4.164
Heppner, Sabrina	sheppner@mail.upb.de	6323	F2.209
Heyi, Binyam Shiferaw	binyam@mail.upb.de	1753	O3.161
Hinnenthal, Kristian	krijan@mail.upb.de	6732	F2.406
Ho, Nam	namh@mail.upb.de	5396	O3.122
Hottung, Andre	ahottenburg@mail.upb.de	26	ZM1.03-10
Hucke, Marion	marion.hucke@upb.de	6695	F2.411
Hüllermeier, Eyke	eyke@upb.de	1771	O4.258
Hülsmann, Adrian	klecks@upb.de	6623	F2.124
Jager, Tibor	tibor.jager@upb.de	6686	F2.308
Jahn, Claudia	claudia.jahn@uni-paderborn.de	6622	F2.104
Jakobs, Marie-Christine	mcjakobs@mai.upb.de	1767	O4.128
Jatzkowski, Jan	jan.jatzkowski@c-lab.de	6115	FU.313
Jazayeri, Bahareh	bahar.jazayeri@upb.de	5465-213	ZM1.03-11
Jovanovikj, Ivan	ijovanovikj@s-lab.upb.de	5465-216	ZM1.03-10
Juhnke, Jakob	jakob.juhnke@upb.de	6328	F2.108
Jung, Daniel	jungd@hni.uni-paderborn.de	6433	F1.125
Karl, Holger	hkarl@mail.uni-paderborn.de	5375	O3.158
Kastens, Uwe	uwe@upb.de	6686	F2.308
Kaufmann, Paul	paul.kaufmann@upb.de	1718	O3.116
Keil, Reinhard	Reinhard.Keil@uni-paderborn.de	6411	F1.428
Keller, Axel	kel@upb.de	1723	O2.149
Kemper, Michaela	mkemper@upb.de	1735	O2.164
Kenter, Tobias	kenter@upb.de	4340	O2.161
Kleine Büning, Hans	kbcs@upb.de	3360	O4.255
Kleinjohann, Bernd	bernd.kleinjohann@c-lab.de	6101	FU.214

Name	E-Mail	Telefon	Raum
Kleinjohann, Lisa	lisa.kleinjohann@c-lab.de	6102	FU.214
Klingler, Florian	klingler@ccs-labs.org	6493	F1.319
Kolb, Christina	ckolb@mail.upb.de	6725	F2.406
König, Jürgen	jkoenig@mail.upb.de	1715	O4.134
Kouchaksaraei, Hadi Razzaghi	hadi.razzaghi.kouchaksaraei@uni-paderborn.de	1758	O3.170
Krämer, Julia-Desirée	juliadk@mail.upb.de	5388	O4.131
Kraus, Hans-Joachim	hajo.kraus@upb.de	6538	F1.410
Krawinkel, Andreas	Krawi@upb.de	1725	O2.155
Kreiss, Lydia	Lydia.Kreiss@uni-paderborn.de	6501	F1.404
Krohn, Jörg-Peter	krohn@upb.de	3325	E1.103
Krüger, Jessica	jessica.krueger@upb.de	6614	F2.114
Krüger, Stefan	stefan.krueger@upb.de	3309	ZM1.02 -12
Kurek, Rafael	rafaeil.kurek@upb.de		
Lange, Christina	chlange@mail.upb.de	1749	O4.213
Lass, Michael	Michael.Lass@upb.de	1722	O2.149
Lengeling, Elisabeth	elisabeth.lengeling@upb.de	3361	O4.155
Lettmann, Theodor	lettmann@upb.de	3350	O4.173
Li, Shouwei	sli@mail.uni-paderborn.de	6468	F1.216
Liske, Gennadij	utyf@mail.upb.de	6626	F2.111
Löken, Nils	nilo@upb.de	6632	F2.111
Lösch, Achim	achim.loesch@upb.de	5395	O3.122
Lücker, Steven Christopher	sluecker@mail.upb.de	6522	F1.107
Mäcker, Alexander	amaecker@mail.uni-paderborn.de	6427	F1.119
Magenheim, Johann S.	jsm@uni-paderborn.de	6341	F2.116
Malatyali, Manuel	malatya@upb.de	6462	F1.122
Maniera, Jürgen	sammy@upb.de	3326	ZM1.02.04
Margaritis, Melanie	melanie.margaritis@upb.de	6603	F2.201
Markarian, Christine	chrisss@mail.upb.de	6460	F1.225
Mehic, Ahmet	amehic@upb.de	3303	ZM1.02.12
Meisner, Sebastian	sebastian.meisner@upb.de	4347	O3.128
Melnikov, Vitaly	melnikov@mail.upb.de	3349	O4.158
Memedi, Agon	memedi@ccss.labs.org	6513	F1.416
Mergard, Claudia	claudia.mergard@upb.de	1727	O2.152
Meyer auf der Heide, F.	fmadh@upb.de	6480	F1.301
Mohr, Felix	fmohr@mail.upb.de	3352	O4.149
Mohsen, Ahmadi Fahander	ahamadim@mail.upb.de	3353	O4.167
	ahamadim@mail.upb.de	6513	F1.416
Monien, Burkhard	bm@upb.de	6707	F2.413
Mory, Martin	martin.mory@upb.de	5013	ZM1.02-13
Müller, Kathrin geb. Bröker	kathyb@uni-paderborn.de	6600	F2.416
Nabeel, Muhammad	nabeel@ccs-labs.org	6515	F 1.412
Neugebauer, Jonas	jonas.neugebauer@uni-paderborn.de	6611	F2.201
Ngonga Ngomo, Acel-Cyrille	axel.ngonga@upb.de	3342	O4.213
Niewöhner, Nadine	nadine.niewoehner@iem.fraunhofer.de	5465-403	ZM1.
Nitsche, Holger	hn@upb.de	1726	O2.155

Name	E-Mail	Telefon	Raum
Nouri, Zahra	znouri@mail.upb.de	5465-207	ZM1.03.07
Oberhoff, Andreas	oberhoff@upb.de	6514	F1.419
Oberhür, Simon	oberthür@upb.de	3844	ZM1.03-08
Ohrndorf, Laura	laura.orndorf@mail.upb.de	6613	F2.201
Parruca, Donald	donald.parruca@upb.de	1754	O3.164
Paul, Adil	adilp@mail.upb.de	3346	O4.161
Peuster, Manuel	manuel.peuster@upb.de	4341	O3.149
Pfahler, Peter	peter@upb.de	6688	F2.311
Pfannschmidt, Karlson	kiudee@mail.upb.de	3346	O4.161
Pierenkemper, Christoph	cpierenk@hni.upb.de		ZM1.
Platenius, Marie Christin	m.platenius@uni-paderborn.de	5465/176	ZM1.02-13
Platzner, Marco	platzner@upb.de	5250	O3.207
Plessl, Christian	christian.plessl@upb.de	5399	O3.110
	christain.plessl@upb.de	5399	O2.167
Podlipyan, Pavel	papo@mail.uni-paderborn.de	6460	F1.216
Rammig, Franz J.	franz@upb.de	6500	F1.414
Riebler, Heinrich	heirich.riebler@upb.de	5382	O2.158
Riechers, Sören	soerenri@mail.upb.de	6427	F1.119
Rittmeier, Florian	florian.rittmeier@s-lab.upb.de	3355	ZM1.03-37
Roger, Irene	irene.roger@upb.de	6620	F2.122
Rose, Sascha			ZM1
Saage, Sonja	ssaage@s-lab.upb.de	5582	ZM1.03-13
Sauer, Stefan	sauer@s-lab@upb.de	5390	ZM1.03-44
Schäfer, Johannes	jschaefer@mail.upb.de	6420	F1.119
Schäfer, Wilhelm	wilhelm@upb.de	2559	B2.326
Schäfermeyer, Petra	petral@upb.de	6481	F1.304
Scharfenbaum, Joachim	joscha@upb.de	3327	E1.101
Scheideler, Christian	scheideler@upb.de	6728	F2.326
Schlatt, Elisabeth	schlatt@mail.upb.de	3764	O4.125
Schneider, Stefan	stefan.schneider@upb.de	1753	O3.170
Schomaker, Gunnar	schomaker@sicp.de	5465-228	ZM1.03-14
Schönberg, Sven	sven.schoenberg@c-lab.de	6121	FU.234
Schroeder, Ulf-Peter	ups@upb.de	6726	F2.409
Schubert, Philipp	phillip.schubert@upb.de	5584	ZM1.02-11
Schulte, Carsten	carsten.schulte@upb.de	6343	F2.116
Schultz-Friese, Tobias	tsf@upb.de	6664	F2.224
Schwabe, Arne	arne.schwabe@upb.de	1756	O3.146
Schwichtenberg, Simon	schwicht@mail.upb.de	5290	ZM1.03-09
Selke, Harald	hase@upb.de	6413	F1.101
Senft, Björn	bjoern.senft@upb.de	5465-211	ZM1.03-08
Setzer, Alexander	asetzer@mail.upb.de	6721	F2.403
Shaker , Ammar	ammar.shaker@upb.de	3345	O4.164
Simon, Jens	simon@upb.de	1731	O2.164
Skopalik, Alexander	skopalik@mail.upb.de	6457	F1.209
Sommer, Christoph	sommer@ccs-labs.org	6495	F1.322
Stahl, Katharina	kasia@upb.de	5465-255	ZM1.03-37
Stoll, Christa	stoll@upb.de	3339	O4.213

Name	E-Mail	Telefon	Raum
Strothmann, Thim	thim@mail.upb.de	6730	F2.323
Szwillus, Gerd	szwillus@upb.de	6624	F2.119
Teetz, Alexander	alexandwer.teetz@upb.de	5465-215	ZM1.03-10
Thissen, Thomas	tici@upb.de	6701	F2.320
Töws, Manuel	mtoews@upb.de	4272	O4.122
Travkin, Oleg	oleg82@upb.de	1709	O4.131
Utermöhle, Michael	mike@upb.de	6666	F2.224
Vaz, Gavin	gavin.vaz@upb.de	1717	O2.158
von Rooijen, Lorijn	lorijn.van.rooijen@uni-paderborn.de	5713	ZM1.03-09
Wahby, Mostafa	mostafa.wahby@uni-paderborn.de	6468	F1.213
Wassing, Heinz-Georg,	koala@upb.de	6430	F1.122
Wawilow, Anastasia	apetkau@upb.de	6625	F2.124
Wegener, Friedhelm	fw@upb.de	3354	ZM1.03-04
Wehrheim, Heike	wehrheim@upb.de	4331	O4.225
Weskamp, Christoph	weskamp@dsor.de	5240	O2.466
Wiechers, Beatrix	wiechers@upb.de	3336	ZM1.03.46
Wiederhold, Cornelia	connyw@upb.de	6523	F1.101
Wiersema, Tobias	tobias.wiersema@upb.de	4343	O3.125
Wilke, Adrian	adrian.wilke@upb.de	6733	F2.416
Winkelkemper, Felix	winfel@hni.upb.de	6410	F1.104
Winnwa, Nils / Azubi	nils.winnwa@uni-paderborn.de	1727	O2.152
Wolters, geb. Bokermann, Dennis	dennis.bokermann@upb.de	5465-221	ZM1.03-07
Ygitbas, Enes	enes@upb.de	3356	ZM1.03-06
Zahn, Gero	gero.zahn@upb.de	6519	F1.107

### 3 Weitere wichtige Adressen

Name	E-Mail	Telefon	Raum
Fachschaft Mathematik/Informatik	fsmi@upb.de	3260	E1.311
Mathe-Treff		3775	D3.331
Mathe-Lernzentrum		1856	J2.324
Prüfungssekretariat Mathematik/CE:			
Ramazan Uysal	zps.math1@zv.upb.de	4230	C2.326
Prüfungssekretariat Informatik:			
Manuel Leßmann	zps.inf1@zv.uni-paderborn.de	5207	C2.332
Rechnerbetreuung Didaktik	intermax@upb.de	3758	D2.339
Rechnerbetrieb Mathematik	pem@math.upb.de	3494	D2.301
Rechnerbetreuung Informatik			
Campus	IRB-Support@upb.de	3322	E1.303
Rechnerbetreuung Informatik			
Fürstenallee	IRB-Support@upb.de	6755	F2.521
Rechnerbetreuung Informatik			
Techniker	IRB-Support@upb.de	3318/4332	E1.303



## 4 Veranstaltungen

### 4.1 Übersicht

Vorlesungen, für die uns bis Redaktionsschluss keine Kommentare erreicht haben, sind in der folgenden Übersicht mit -- gekennzeichnet.

## Vorlesungen der Informatik

### Grundstudium Bachelorstudiengang

Sommer	Programmierung	(1.1)	13
Brunthaler	Programmiersprachen	(1.1)	--
Sauer	Softwareentwurf	(1.2)	--
Wehrheim, Hüllermeier	Modellierung	(2.1)	14
Blömer	Einführung in Berechenbarkeit, Komplexität und Formale Sprachen	(2.3)	16
Hellebrand	Grundlagen der Rechnerarchitektur	(3.1)	18
Jäger	IT-Sicherheit		--

### Hauptstudium Bachelorstudiengang

#### Algorithmen und Komplexität

Scheideler	Fundamental Algorithms	(2.1)	19
Meyer auf der Heide	Parallelität und Kommunikation	(2.1)	21

#### Computersysteme

Dressler	Networked Embedded Systems	(3.1)	22
Dressler	Operating Systems	(3.1)	23
Dressler	History of Computer Systems	(3.1)	24

#### Daten und Wissen

Domik	Computer Graphics Rendering	(4.1)	--
Böttcher	Datenbanken und Informationssysteme	(1.1)	--
Ngonga	Knowledge Graphs	(1.1)	--

#### Softwaretechnik

Pfahler	Programming Languages and Compilers	(1.1)	26
Bodden	Secure Software Engineering	(1.1)	27
Szwillus	Interaktionsgestaltung	(4.1)	28
Engels	Advanced Software Engineering: Methods, Architectures, Industrial Applications	(4.1)	39

## Proseminare

Weiland	Praktische Umsetzung der barrierefreien Gestaltung	--
Scheideler	Effiziente Algorithmen	29

## Masterstudiengang

### Algorithm Design

Skopalik	Algorithmic Game Theory	(2.1,2.2)	30
Meyer auf der Heide	Routing and Data Management in Networks	(2.1,2.2,2.4)	33
Fischer	Algorithms for Highly Complex Virtual Scenes	(2.1,2.2)	31

### Computer Systems

Platzner	Advanced Computer Architecture	(3.4,3.5,3.6)	--
Platzner	Reconfigurable Computing	(3.4,3.5)	--
Plessl	High-Performance Computing	(3.1,3.2,3.5)	--

### Intelligence and Data

Lettmann	Planning and Heuristic Search	(2.1,2.2)	--
Hüllermeier	Machine Learning 1	(1.4)	32

### Networks and Communication

Meyer auf der Heide	Routing and Data Management in Networks	(2.1,2.2,2.4)	--
Karl	Mobile communication	(3.1,3.3)	34

### Software Engineering

Wehrheim	Software Analysis	(1.1,1.5)	35
Tauber	Extra-Ordinary Human-Computer Interaction, Accessibility	(4.2,4.3,4.5)	--
Anjorin	Fundamentals of Model-Driven Engineering	(1.1,1.2,1.5,1.6)	37
Selke	Kontextuelle Informatik	(4.2)	36
Plessl	High-Performance Computing	(3.1,3.2,3.5)	--
Hermann	Type Systems for Correctness and Security	(1.2, 1.3, 1.5, 1.6)	40

## Seminare

Wehrheim	Software Transactional Memory	(1.1,1.5,1.6)	41
Bodden	Secure Systems Engineering	(1.1,1.2,1.5,1.6)	--
Meyer auf der Heide	Gems of Theoretical Computer Science	(2.1-2.4)	42
Selke	Digital Humanities	(4.1-4.7)	43

Scheideler	PRISMA	(2.1,2.2)	--
Dressler	Selected Topics in Wireless Networking	(3.1-3.6)	44

## Projektgruppen

Projektgruppen die bereits im letzten Semester gestartet sind, werden nicht mit aufgeführt.

Bodden	PG Secure Integration of Cryptographic Software		45
--------	---	--	----

## Oberseminare

Böttcher	Datenbanken	(SWT&IS)	--
Hüllermeier	Maschinelles Lernen	(SWT&IS)	--
Wehrheim	Spezifikation und Modellierung	(SWT&IS)	--
Kleine Büning	Wissensbasierte Systeme	(SWT&IS)	--
Bodden	Softwaretechnik	(SWT&IS)	--
Engels	Informationssysteme	(SWT&IS)	--
Ngonga	Semantic Web	(SWT&IS)	--

Scheideler	Theoretische Informatik	(MuA)	??
Meyer auf der Heide	Algorithmen und Komplexität	(MuA)	46
Blömer	Codes & Kryptographie	(MuA)	--
Meyer auf der Heide	SFB 901 On-The-Fly Computing	(MuA)	??
Domik	Interactive 3D Graphics and Games	(MuA/MMWW)	--

Jager	Computersicherheit	(ESS)	--
Karl	Rechnernetze	(ESS)	??
Platzner	Technische Informatik	(ESS)	--
Dressler	Distributed Embedded Systems	(ESS)	--
Platzner	Approximate Computing	(ESS)	--
Plessl	Hochleistungs-IT-Systeme	(ESS)	--

Schulte	Informatik und Bildung	(MMWW)	47
Keil	Informatik und Gesellschaft	(MMWW)	--
Domik	Computergrafik, Visualisierung und Bildverarbeitung	(MMWW)	--
Szwillus	Mensch-Computer-Interaktion	(MMWW)	--

Schulte	Stufenbezogene Unterrichtsmodelle		48
Schulte	Fachdidaktische Grundlagen		49

Schulte	Fachdidaktische Konzepte	50
---------	--------------------------	----

### **Lehrveranstaltungen für andere Studiengänge**

Fischer	Grundlagen der Programmierung für Ingenieure II	51
Fischer	Grundlagen der Programmierung für MB	52
Pfahler	Einführung in die Web-bezogenen Sprachen	53
Fischer	Datenverarbeitung	54
Karl	Projektmanagement für Computer Engineering	55

### **Allgemeine Veranstaltungen**

Gerstenberger	Tutorenschulung Informatik	--
Gerstenberger	Agile Organisation von Lerngruppen	57
Gerstenberger	Evaluation des Workloads von Lerngruppen	56
Plessl	PC <sup>2</sup> Kolloquium: Kolloquium der Parallelverarbeitung	--

## 4.2 Informatik

### Programmierung

**Dozent:** Sommer

**Büro:** F1.322

**Sprechstunde:** nach Absprache

#### Inhaltsangabe

Das Entwickeln von Software ist ein zentraler Tätigkeitsbereich in der Informatik. Software-Entwickler müssen Aufgaben analysieren und modellieren, Software-Strukturen entwerfen und diese in einer Programmiersprache implementieren können.

Die Vorlesung „Grundlagen der Programmierung“ soll dazu einen Einstieg in systematisches Programmieren liefern. Wir besprechen unterschiedliche Programmierparadigmen und -sprachen, das Herangehen an ein Programmierproblem, geeignete Werkzeuge und Techniken.

#### Verschiedenes

**Hörerkreis:**

Bachelor Informatik

**Prüfungsform:**

Klausur

**Homepage:**

<http://www.ccs-labs.org/teaching/gp/2017w/>

# Modellierung

**Dozent:** Hüllermeier, Wehrheim

**Büro:** O4.258

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Das Modul Modellierung ist unter der Nummer I.2.1. im Modulhandbuch für den Bachelor Informatik detailliert in Bezug auf seine Rolle innerhalb des Studiengangs, auf die Inhalte und die Lernziele beschrieben. Es ist ein Pflichtmodul in den Bachelorstudiengängen Informatik und Wirtschaftsinformatik (1. Semester) und Pflichtveranstaltung im Studiengang Ingenieurinformatik und im Lehramtsstudium (3. Semester) und im Diplomstudiengang der Medienwissenschaften mit Schwerpunkt Medieninformatik.

### Inhaltliche Gliederung der Veranstaltung:

1. Einführung: Begriffe Modell, Modellierung
2. Grundlegende Formalismen: Mengen, Relationen, Funktionen, induktive Definitionen und Beweise
3. Logik: Aussagenlogik, Prädikatenlogik, Kalküle, Modellierung und Beweise
4. Grundlagen der Kombinatorik und diskreten Wahrscheinlichkeitsrechnung
5. Modellierung mit Graphen: Weg, Verbindung, Zuordnung, Abhängigkeiten, Abfolgen, Fluss
6. Modellierung von Strukturen: reguläre Ausdrücke und kontextfreie Grammatiken
7. Modellierung von Abläufen: endliche Automaten

**Umfang und Zeiten:** Der Arbeitsaufwand einschließlich der Prüfungsleistungen ist mit 8 Leistungspunkten festgelegt und entspricht durchschnittlich 240 Stunden individueller Arbeitszeit. Präsenzstunden sind 4 Vorlesungsstunden (V4) und 2 Übungsstunden (Ü2).

**Dozenten:** Prof. Hüllermeier und Prof. Wehrheim halten die Vorlesung gemeinsam.

## Literaturangaben

- **Kastens, Kleine Büning:** *Modellierung*, Hanser Verlag, 2. Auflage, 2008

## Verschiedenes

### Hörerkreis:

Bachelor (Ing.-) Informatik, Bachelor Wirtschaftsinformatik, Lehramt Informatik, Diplom Medienwissenschaften (Medieninformatik)

### Modulzugehörigkeit:

BA Informatik: Gebiet I.2 Modelle und Algorithmen

**Prüfungsform:**

Klausur, Bekanntgabe zusätzlicher Modalitäten zu Beginn des Semesters

**Leistungspunkte:**

8

**vorausgesetzte Kenntnisse:**

Oberstufenstoff Mathematik, Bereitschaft und Fähigkeit zum Erlernen formaler Kalküle

**weiterführende Veranstaltungen:**

Datenstrukturen und Algorithmen

**nächster Wiederholungstermin:**

WS 2018/19

# Einführung in Berechenbarkeit, Komplexität und formale Sprachen

**Dozent:** Blömer

**Büro:** F2.101

**Sprechstunde:** Mi, 13 - 14 Uhr

## Inhaltsangabe

Was können Computer berechnen? Sind alle Rechner gleichmächtig? Wie packt man ein Auto? Wie schwierig sind Kreuzworträtsel? Wie analysiert man Sätze, die von Personen, welche hier (soll heißen jetzt) ungenannt bleiben, kommen, die dazu neigen, Sätze immer tiefer zu schachteln? Warum können Automaten nicht zählen, aber die Teilbarkeit durch sieben feststellen? Das sind alles Fragen, die im Rahmen dieser Vorlesung beantwortet werden.

Die Theoretische Informatik ist das Fundament der Informatik und sie bildet ein wertvolles Basiswissen in der Arbeit jedes Informatikers.

Wir werden unter anderem folgende Teilgebiete kennen lernen:

- Berechenbarkeit
  - Turingmaschinen
  - entscheidbare und rekursiv aufzählbare Sprachen
  - nicht entscheidbare Probleme
  - Halteproblem
  - nicht rekursiv aufzählbare Probleme
- Komplexitätstheorie
  - Klassen P und NP, Zeitkomplexität
  - NP-Vollständigkeit
  - Satz von Cook
  - Reduktion
- Algorithmen: Behandlung NP-vollständiger Probleme
  - Heuristiken: Backtracking, Branch and Bound
  - Approximationsalgorithmen



- Formale Sprachen
  - reguläre Sprachen, reguläre Grammatiken
  - deterministische und nicht-deterministische Automaten
  - reguläre Ausdrücke
  - Äquivalenzsatz
  - Pumping Lemma
  - kontextfreie Sprachen, kontextfreie Grammatiken
  - Kellerautomaten
  - Chomsky Normalform, Äquivalenzsatz
  - CYK-Algorithmus

### Literaturangaben

- **Michael Sipser:** *Introduction to the Theory of Computation*, PWS, 1997
- **John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman:** *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, 2. Auflage 2001
- **Uwe Schöning:** *Theoretische Informatik - kurzgefasst*, Spektrum, akad. Verlag, Heidelberg, 3. Auflage 1997
- **Ingo Wegener:** *Theoretische Informatik - Eine algorithmenorientierte Einführung*, Teubner, 1993
- **Jon Kleinberg, Éva Tardos:** *Algorithm Design*, Addison Wesley, 2006
- **Juraj Hromkovic:** *Theoretische Informatik, Berechenbarkeit, Komplexitätstheorie, Algorithmik, Kryptographie. Eine Einführung*, Teubner, 2. Auflage 2004

### Verschiedenes

**Hörerkreis:**

Informatik Bachelor, Mathematik Bachelor  
mit Nebenfach Informatik

**Modulzugehörigkeit:**

MuA Modul I.2.3

**Prüfungsform:**

Klausur

**vorausgesetzte Kenntnisse:**

Veranstaltungen der ersten beiden Semester  
des Informatik-Studiums

**nächster Wiederholungstermin:**

jährlich

# Rechnerarchitektur

**Dozent:** Hellebrand

**Büro:** P1.6.08.1

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Die Vorlesung „Rechnerarchitektur“ vermittelt die Grundlagen für den Aufbau moderner Rechner-systeme. Als durchgängiges Beispiel wird der MIPS Prozessor herangezogen. Die Schwerpunkte der Lehrveranstaltung sind:

- Grundstrukturen, von Neumann Rechner
- Leistungsbewertung
- Befehlssätze und Assemblerprogrammierung
- Datenpfad und Steuerung
- Pipelining
- Speicherhierarchie, insb. Cache-Management und virtueller Speicher
- Ein-/Ausgabe

Die Übungen bestehen aus theoretischen und praktischen Aufgaben, die von den Studierenden selbstständig gelöst werden sollen. In den Übungsstunden werden die Lösungen besprochen.

## Literaturangaben

- **D. A. Patterson, J. L. Hennessy:** *Computer Organization & Design – The Hardware / Software Interface (5th Edition)*, Morgan Kaufmann, 2010

## Verschiedenes

**Hörerkreis:**

Bachelor Informatik, Computer Engineering,  
Elektrotechnik

**Prüfungsform:**

Klausur

**Leistungspunkte:**

5

**vorausgesetzte Kenntnisse:**

Grundlagen der Technischen Informatik  
(GTI)

**weiterführende Veranstaltungen:**

Wahlpflichtveranstaltungen aus “Computer-  
systeme”

**nächster Wiederholungstermin:**

WS 2018/19

**Homepage:**

[http://www.date.upb.de/pages/de/  
lehre.php](http://www.date.upb.de/pages/de/lehre.php)

# Grundlegende Algorithmen (Fundamental Algorithms)

**Dozent:** Scheideler

**Büro:** F2.326

**Sprechstunde:** Do, 16-17 Uhr

## Inhaltsangabe

Die Vorlesung wird sich mit folgenden Themen beschäftigen:

1. Fortgeschrittene Datenstrukturen
  - Binomial Heaps und Fibonacci Heaps
  - Fortgeschrittene Hashingverfahren
2. Graphalgorithmen
  - Zusammenhangskomponenten
  - Kürzeste Wege
  - Matchings
3. Netzwerkfluss
  - Grundlagen
  - Ford-Fulkerson Algorithmus
  - Preflow-Push Algorithmus
  - Mehrgüterflüsse
4. Stringmatching
  - Knuth-Morris-Pratt Algorithmus
  - Boyer-Moore Algorithmus
  - Aho-Corasick Algorithmus

## Literaturangaben

- **T.H. Corman and C.E. Leiserson and R.L. Rivest:** *Introduction to Algorithms*, MIT Press, 1990
- **R.K. Ahuja and T.L. Magnanti and J.B. Orlin:** *Network flows*, Prentice Hall, 1993
- **G.A. Stephen:** *String searching algorithms*, World Scientific Publishing, 1994
- **S.O. Krumke, H. Noltemeier:** *Graphentheoretische Konzepte und Algorithmen*, Teubner, 2005
- **A.V. Aho and J.E. Hopcroft and J.D. Ullman:** *Data Structures and Algorithms*, Addison-Wesley, 1987

- **M.R. Garey and D.S. Johnson:** *Computers and intractability*, Freeman, 2000

### Verschiedenes

**Hörerkreis:**

Informatik Bachelor

**Prüfungsform:**

Klausur

**weiterführende Veranstaltungen:**

Methoden des Algorithmenentwurfs

**Homepage:**

<http://cs.uni-paderborn.de/ti/lehre/veranstaltungen/ws-2017/ga.html>

**Modulzugehörigkeit:**

Info 2. Studienabschnitt, MuA

**vorausgesetzte Kenntnisse:**

Datenstrukturen und Algorithmen

**nächster Wiederholungstermin:**

WS 2018

# Parallelität und Kommunikation

**Dozent:** Meyer auf der Heide

**Büro:** F1.301

**Sprechstunde:** Mi, 13-14 Uhr

## Inhaltsangabe

Die Vorlesung beschäftigt sich mit effizienten Methoden, Kommunikation zwischen Prozessoren eines Parallelrechners zu realisieren. Die Hardware großer Parallelrechner stellt hierfür typischerweise ein Netzwerk zur Verfügung durch das die zu kommunizierenden Daten laufen. In der Vorlesung stellen wir Methoden vor, Kommunikation durch Routing im Netzwerk, durch Simulation des Kommunikationsgraphen auf dem Netzwerk und mit Hilfe globaler Variablen zu realisieren.

Es werden insbesondere folgende Themen behandelt:

- Permutationsrouting auf Gittern
- Sortiernetzwerke
- Oblivious Routing und probabilistisches Routing im Butterfly-Netzwerk
- Datenverwaltung in Netzwerken

## Literaturangaben

- **Frank Thomson Leighton:** *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, 1992
- **Ian Parberry:** *Parallel Complexity Theory*, Pitman/Wiley, 1987

## Verschiedenes

**Hörerkreis:**

Bachelor Informatik

**Modulzugehörigkeit:**

2. SA Bachelors, MuA Modul II.2.1

**Prüfungsform:**

voraussichtlich mündliche Prüfung

**vorausgesetzte Kenntnisse:**

Datenstrukturen und Algorithmen

**Homepage:**

<http://www.hni.upb.de/en/alg/teaching>

# Networked Embedded Systems (NES)

**Dozent:** Dressler

**Büro:** F1.401

**Sprechstunde:** nach Absprache

## Inhaltsangabe

Ziel des Kurses ist es, ein tiefgehendes Verständnis in den Aufbau und die Programmierung vernetzter eingebetteter Systeme im Allgemeinen und in Sensornetzen im Speziellen zu vermitteln. Dazu werden in der Vorlesung entsprechende Grundlagen von Sensornetzen diskutiert, die dann im Rahmen der Übung praktisch umgesetzt werden sollen.

- Aufbau eines Sensorknotens  
Architektur eingebetteter Systeme, Programmierparadigmen
- Sensornetze  
Verknüpfung mehrerer Knoten zu einem Netz
- Routing  
Einfache Routingalgorithmen
- Lokalisierung  
Lokalisierungsmethoden, Signalstärke, Laufzeitmessungen

## Literaturangaben

- **Falko Dressler:** *Self-Organization in Sensor and Actor Networks*, Chichester, John Wiley & Sons, 2007
- **Holger Karl and Andreas Willig:** *Protocols and Architectures for Wireless Sensor Networks*, Chichester, John Wiley & Sons, 2005

## Verschiedenes

**Hörerkreis:**  
Master Informatik, Master CE

**Prüfungsform:**  
mündlich

**vorausgesetzte Kenntnisse:**  
KMS, Rechnernetze

**nächster Wiederholungstermin:**  
WS 2018/2019

**Homepage:**  
<http://www.ccs-labs.org/teaching/nes/>

# Operating Systems

**Dozent:** Dressler

**Büro:** F1.401

**Sprechstunde:** nach Absprache

## Inhaltsangabe

In this class, we discuss general aspects of operating systems as well as specific characteristics of real-time operating systems and operating systems for embedded systems.

Topics of interest include:

- Parallelism
- Scheduling
- Synchronization
- Inter-Process Communication
- Memory Management
- Security
- Embedded OS
- Real-Time

## Literaturangaben

- **Andrew S. Tanenbaum:** *Modern Operating Systems*, ed. 3rd, Upper Saddle River, NJ, Pearson-Prentice Hall, 2009
- **William Stallings:** *Operating Systems: Internals and Design Principles*, ed. 8th, Upper Saddle River, Pearson Prentice Hall, 2014
- **Abraham Silberschatz, Peter B. Galvin and Greg Gagne:** *Operating Systems Concepts*, ed. 9th, John Wiley & Sons, 2013

## Verschiedenes

**Hörerkreis:**

Bachelor Informatik, Bachelor Computer Engineering

**nächster Wiederholungstermin:**

Winter 2018/2019

**Homepage:**

<http://www.ccs-labs.org/teaching/os/>

# History of Computer Systems

**Dozent:** Dressler

**Büro:** F1.401

**Sprechstunde:** nach Absprache

## Inhaltsangabe

Ziel dieser Vorlesung ist es, die Entstehungsgeschichte der Informationstechnik kennenzulernen und zu verstehen. Dabei sollen nicht nur die historischen Begebenheiten und Zusammenhänge vorgestellt, sondern auch die Funktionsweise der Maschinen und Computer erklärt werden. Die Frage, warum heutige Computer so gebaut sind und so programmiert werden, wie man es im Informatikstudium lernt, führt dabei als roter Faden durch die Veranstaltung und soll anhand der historischen Entwicklung geklärt werden. Die Inhalte der Vorlesung werden durch Übungen vertieft, in denen Studierende selbst historische Computer (teils mit echten Maschinen, teils mit Simulationen) bedienen sollen.

Ein Besuch im Heinz Nixdorf MuseumsForum rundet die Veranstaltung ab.

Die Inhalte des Kurses umfassen:

- Vor dem Computer: Rechenhilfen und Rechenmaschinen
- Die ersten Computer in Deutschland, Großbritannien und den USA 1938-1948
- Die theoretischen Fundamente der Informatik
- Speichertechniken
- Rechentechniken
- Analogcomputer
- Bürocomputer und Großrechner in den 60ern bis 80ern
- Geschichte der Kommunikationstechnik
- Programmiersprachen
- Personal Computer und Computerspiele

## Verschiedenes

**Hörerkreis:**

Bachelor Informatik, Bachelor Computer Engineering

**Prüfungsform:**

Klausur

**nächster Wiederholungstermin:**

Winter 2018/2019

**Homepage:**

<http://www.ccs-labs.org/teaching/hoc/>



# Rechnernetze

**Dozent:** Karl

**Büro:** O3.158

**Sprechstunde:** Montag, 13 - 14 Uhr

## Inhaltsangabe

Die Veranstaltung beschreibt die Grundlagen, Architekturen und Protokolle der Rechnernetze und Telekommunikationsnetze, mit besonderem Schwerpunkt auf Internet-basierten Ansätzen. Die Veranstaltung vertieft die Vorkenntnisse aus KMS. Wesentliche Themen der Veranstaltung sind:

- Grundlegende Abstraktionen - Protokoll, Service, Layering, Multiplexing, Scheduling, Duplex, ...
- Architekturmodelle: ISO/OSI, Internet
- Programmierschnittstelle
- Grundlagen der Nachrichtentheorie, Eigenschaften der physikalischen Übertragung
- Vielfachmedienzugriff (Medium Access Control) und Sicherungsschicht
- Wegewahl, Routing, Forwarding; Netzstrukturen
- Überlastabwehr, Flußkontrolle.

Mehr über die Inhalte erfahren Sie auch in den Foliensätzen der ersten beiden Kapitel.

## Literaturangaben

- **A. Tanenbaum:** *Computer Networks*, 4. Auflage, Prentice Hall
- Alternativ:
- **L. L. Peterson & B. S. Davie:** *Computer Networks – A Systems Approach*, 2003, 3rd edition, Morgan Kaufman
  - **J. F. Kurose & K. W. Ross:** *Computer Networking: A Top-Down Approach Featuring the Internet*, 2004, 3rd edition, Addison Wesley

## Verschiedenes

**Hörerkreis:**

Informatik Bachelor

**Prüfungsform:**

Klausur

**nächster Wiederholungstermin:**

WS 2017/2018

**Modulzugehörigkeit:**

ESS

**vorausgesetzte Kenntnisse:**

KMS oder vergleichbar

**Homepage:**

<https://cs.uni-paderborn.de/cn/teaching/courses/rechnernetze/>

# Programming Languages and Compilers

**Dozent:** Brunthaler, Pfahler

**Büro:** O3.149, F2.311

**Sprechstunde:** Nach Vereinbarung

## Inhaltsangabe

Diese Veranstaltung vermittelt Kenntnisse und Fähigkeiten zum vertieften Verständnis, zur Spezifikation und zur Implementierung von Programmier- und Spezifikationssprachen. Die Teilnehmer werden befähigt, grundlegende Kalküle zur präzisen Beschreibung von Spracheigenschaften einzusetzen und Methoden zur Implementierung von Sprachen anzuwenden.

Inhalt:

1. Ebenen von Spracheigenschaften und Struktur von Übersetzern
2. Spezifikation von Grundsymbolen und lexikalische Analyse
3. Syntaktische Spezifikation und Analyse
4. Semantische Eigenschaften und Analyse
5. Transformation in Zielsprachen
6. Zwischensprachen
7. Speicherorganisation
8. Codeerzeugung

Die Veranstaltung hat einen Umfang von 5 SWS und besteht aus Vorlesungsteil und den praktischen Übungen, wo wir Übersetzer-erzeugende Werkzeuge zur Implementierung von Sprachen einsetzen. Die Vorlesung wird auf Englisch gehalten.

## Verschiedenes

**Hörerkreis:**

Informatik Bachelor, 2. Studienabschnitt

**Modulzugehörigkeit:**

Softwaretechnik

**Prüfungsform:**

Klausur

**Leistungspunkte:**

6

**nächster Wiederholungstermin:**

WS 2018/19

**Homepage:**

Das Veranstaltungsmaterial wird voraussichtlich in koALA bereitgestellt

# Secure Software Engineering

**Dozent:** Bodden

**Büro:** ZM1.02-09

**Sprechstunde:** Nach Absprache

## Inhaltsangabe

What does it take to engineer software systems securely? This is the key question we wish to address in this course. Answering it requires to develop an understanding of the following key areas of secure software engineering: threat modeling, secure design, secure coding, security validation, secure deployment and maintenance. In this course we will be covering those areas in an example-driven style, discussing current techniques applicable to those areas and lessons learned from concrete real-world security breaches.

In each lecture we will (1) discuss one or two „Vulnerabilities of the day“ and (2) discuss crucial elements of a Secure Software Development Lifecycle, including:

- Threat modeling
- Risk analysis
- Architectural security
- Secure coding
- Secure configuration and deployment
- Updates and maintenance

## Literaturangaben

We will not be able to provide a script for this course. However, a lot of the topics are also covered in the book:

- **Gary McGraw:** *Software Security: Building Security*, which you can access within the university network

## Verschiedenes

**Hörerkreis:**

Informatik Bachelor

**vorausgesetzte Kenntnisse:**

Participants are expected to have completed or nearly completed the first section of the Bachelor degree. On top of that there are no special prerequisites for this course.

**nächster Wiederholungstermin:**

Jedes Wintersemester

**Prüfungsform:**

Klausur

**weiterführende Veranstaltungen:**

Master Courses „Build It, Break It, Fix It“ and „Designing code analyses for large-scale software systems (DECA)“

**Homepage:**

<https://www.hni.uni-paderborn.de/swt/lehre/sse/>

# Interaktionsgestaltung

**Dozent:** Szwillus

**Büro:** F2.119

**Sprechstunde:** siehe Webseite

## Inhaltsangabe

Die Vorlesung behandelt die Gestaltung von Interaktion mit klassischen, graphischen Desktopsystemen genauso wie die Interaktionsgestaltung im Web. Dabei baut sie auf der Pflichtvorlesung „Gestaltung von Nutzungsschnittstellen“ auf, in der erste Grundkenntnisse des Usability Engineering vermittelt wurden. Hier werden die Inhalte vertieft und besonders intensiv die existierenden Usability-Regeln, sowie die darauf aufbauenden Verfahren behandelt und auch praktisch eingeübt. Es wird anschließend deutlich gemacht, was sich ändert, wenn wir statt klassischer graphischer Oberflächen die Situation bei der Gestaltung von Webauftritten betrachten. Die erheblichen existierenden Gemeinsamkeiten beider Bereiche, aber auch ihre Abgrenzungen werden deutlich gemacht. Im Web tritt an die Stelle des Dialogdesigns das Navigationsdesign, was alternative Entwicklungsstrategien erfordert.

Besondere Betonung wird in der Vorlesung auf die stets wachsende Nutzung mobiler Geräte mit ihrer andersartigen technischen Ausrüstung gelegt. Dadurch und durch den sehr anderen Nutzungskontext entstehen neue Benutzungsanforderungen, die durch neuartige Entwicklungsprozesse und -techniken abgefangen werden müssen. Um das Thema abzurunden, wird die bedeutende Rolle von Farbe und Typographie im Web thematisiert.

## Literaturangaben

- **Ben Shneiderman:** *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2009
- **David Benyon:** *Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design*, 2009
- **Jakob Nielsen und Raluca Budiu:** *Mobile Usability*, 2012

## Verschiedenes

**Hörerkreis:**  
Informatik Bachelor

**Prüfungsform:**  
Klausur

**vorausgesetzte Kenntnisse:**  
Die Vorlesung baut auf der vorangegangenen Pflichtvorlesung Gestaltung von Nutzungsschnittstellen auf, in der bereits grundlegende Inhalte des Usability Engineering eingeführt werden.

**Modulzugehörigkeit:**  
Wahlpflichtmodul Interaktionsgestaltung /  
Interaction Design (neue PO)

**Leistungspunkte:**  
6

**Homepage:**  
<https://cs.uni-paderborn.de/mci/lehre/veranstaltungen/ws-2017-2018>

## Proseminar: Effiziente Algorithmen

**Dozent:** Scheideler

**Büro:** F2.326

**Sprechstunde:** Do, 16-17 Uhr

### Inhaltsangabe

Es gibt bereits seit vielen Jahren einen renommierten internationalen Programmierwettbewerb für Studenten, den International Collegiate Programming Contest (ICPC). Ziel des Proseminars ist es, die Studierenden mit den Problemstellungen im ICPC vertraut zu machen und ihnen beizubringen, wie effiziente Verfahren für diese Problemstellungen entwickelt werden können. Dazu wird es wöchentliche Treffen geben, in denen in Gruppenarbeit Probleme besprochen und geeignete Verfahren dafür entwickelt werden. Als wöchentliche Hausaufgabe werden ausgewählte Verfahren anschließend implementiert und eingereicht. Den besten Teams wird bei Interesse die Möglichkeit gegeben, kostenfrei an einem ICPC-Wettbewerb teilzunehmen.

Die Proseminarnote wird sich daran orientieren, ob an den Treffen regelmäßig und aktiv teilgenommen wird und wie gut die Probleme (nach einer Einarbeitungszeit) gelöst werden können.

### Literaturangaben

- **Steven Skiena and Miguel Revilla:** *Programming Challenges - The Programming Contest Training Manual*, Springer Verlag 2003

### Verschiedenes

**Hörerkreis:**

Informatik Bachelor, Wirtschaftsinformatik

**Modulzugehörigkeit:**

MUA

**Prüfungsform:**

Erfolgreiche Teilnahme

**vorausgesetzte Kenntnisse:**

Datenstrukturen und Algorithmen  
Hilfreich: Grundlegende Algorithmen

**nächster Wiederholungstermin:**

WS 2018

**Vorbereitung:**

Wird noch bekannt gegeben

**Homepage:**

<http://cs.uni-paderborn.de/ti/ws-2017/ea.html>

# Algorithmic Game Theory (in english)

**Dozent:** Skopalik

**Büro:** F1

## Inhaltsangabe

In algorithmic game theory we analytically study scenarios involving the interaction of rational agents, such as traffic systems with selfish agents (e.g., cars) routing through a network. Typical questions concern the incentives in such environments - will the agents agree on a common routing pattern? Is there more than one such stable outcome? Is it „fair“ or „efficient“? Will the dynamics evolving from selfish behavior converge and how long does this take? Another prominent issue are efficient algorithms for computing stable allocations in systems with selfish agents, e.g., when matching buyers to sellers, goods to consumers, or partners to each other. Such algorithms use information and preferences reported by agents, and agents might want to manipulate the algorithm by misreporting preferences to get a better good/partner/etc. The challenge here is not only to design algorithms with socially favorable outcomes, but also to manage the incentives using, e.g., payments or threats of punishment, to discourage agent manipulation. Traditionally, techniques from microeconomics are used for mathematical formulation and solution of such problems. Recently, however, these problems are arising within computational environments - most prominently on the Internet and in e-commerce - and here ideas from game theory must be combined with approximation algorithms, distributed computing, and computational complexity.

## Verschiedenes

**Hörerkreis:**

Informatik

**Modulzugehörigkeit:**

MuA III.2.1, III.2.2, III.2.4

**Prüfungsform:**

mündl. Prüfung

**Leistungspunkte:**

5

## Algorithms for highly complex virtual scenes

**Dozent:** Fischer

**Büro:** F1.223

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

Walkthrough systems allow viewing and a walk through virtual 3D scenes. Such systems are used in architectural applications, simulations or rendering of CAD data. The efficiency of real-time rendering algorithms is crucial for a smooth and fast rendering of virtual 3D scenes in a walkthrough system. There are various approaches to simplify highly complex 3D geometric data and to achieve a rendering of the data in real time. In the lecture, we will learn algorithmic approaches in the areas of visibility culling, simplification, level of detail, point based rendering, image-based and parallel rendering.

### Literaturangaben

- **Tomas Akenine-Möller, Eric Haines:** *Real-Time Rendering*, AK Peters, 2002
- **David Luebke, Martin Reddy, Jonathan D. Cohen:** *Level of Detail for 3D Graphics*, Morgan Kaufmann Publishers, 2002
- **Thomas Rauber:** *Algorithmen in der Computergraphik*, Teubner, 1993

### Verschiedenes

**Hörerkreis:**

Master

**Modulzugehörigkeit:**

MuA Modul III 2.1, 2.2

**Prüfungsform:**

Oral examination

**Leistungspunkte:**

6

**vorausgesetzte Kenntnisse:**

Basic data structures and algorithms of Bachelor studies.

**weiterführende Veranstaltungen:**

Master Thesis

**nächster Wiederholungstermin:**

Winter term 2018/19

**Homepage:**

<http://www.hni.upb.de/alg/lehre>

# Machine Learning I (in English)

**Dozent:** Hüllermeier

**Büro:** O4.258

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Due to the ever increasing amount of data that is routinely produced in our information society, the topic of machine learning has become increasingly important in the recent years, not only as a scientific discipline but also as a key technology of modern software and intelligent systems. This lecture provides an introduction to the topic of machine learning, with a specific focus on supervised learning for classification and regression. The lecture covers theoretical foundations of generalisation as well as practical topics and concrete learning algorithms.

1. Introduction
2. The Learning Problem
3. Training versus Testing
4. The Linear Model
5. Non-Linear Methods
6. Overfitting

## Literaturangaben

- **Y.S. Abu-Mostafa, M. Magdon-Ismail:** *H.T. Lin. Learning from Data*, AMLBook, 2012
- **P. Flach:** *Machine Learning*, Cambridge Univ. Press, 2012
- **E. Alpaydin:** *Machine Learning*, Oldenbourg, 2008
- **C.M. Bishop:** *Pattern Recognition and Machine Learning*, Springer, 2006

## Verschiedenes

**Hörerkreis:**

Master Informatik, Mathematik, Wirtschaftsinformatik

**Modulzugehörigkeit:**

M.079.3107, M.079.3114, M.079.8532, M.079.8562, M.079.8566

**Prüfungsform:**

Mündliche Prüfung

**Leistungspunkte:**

6

**vorausgesetzte Kenntnisse:**

Basic knowledge in mathematics (linear algebra, statistics), programming and algorithms.

**weiterführende Veranstaltungen:**

Machine Learning II

**nächster Wiederholungstermin:**

Wintersemester 2018



# Routing and Data Management in Networks

**Dozent:** Meyer auf der Heide

**Büro:** F1.301

**Sprechstunde:** Mi, 13-14 Uhr

## Inhaltsangabe

Routing and data management are basic tasks to be solved in order to allow efficient use of large networks like, e.g., the Internet, peer-to-peer systems, or wireless mobile ad-hoc networks. This lecture deals with algorithms and their analysis for routing and data management in such systems, and especially describes methods for handling their dynamics (movement of nodes, joins and leaves of nodes).

## Literaturangaben

Can be found on the website.

## Verschiedenes

**Hörerkreis:**

Master Informatik

**Prüfungsform:**

oral exam

**Homepage:**

<http://www.hni.upb.de/en/alg/teaching>

**Modulzugehörigkeit:**

MuA 3.SA Modul III.2.1, III.2.2, III.2.4

**vorausgesetzte Kenntnisse:**

Bachelor

# Mobilkommunikation

**Dozent:** Karl

**Büro:** O3.158

**Sprechstunde:** Montag, 13 - 14 Uhr

## Inhaltsangabe

Die Vorlesung gibt einen Überblick über Systeme, Architekturen und Protokolle zur drahtlosen und mobilen Kommunikation. Behandelt werden unter anderem:

- Drahtlose Kommunikation, drahtlose Kanäle
- Medienzugriff in drahtlosen Medien
- Drahtlose lokale Netze (WLAN, IEEE 802.11)
- Systeme der zellularen Mobilkommunikation (GSM, UMTS)

## Literaturangaben

- **J. Schiller:** *Mobilkommunikation*, dpunkt
- **J. Schiller:** *Mobile Communications*, Addison Wesley, derzeit 2. Auflage, 3. Auflage bald erhältlich

## Verschiedenes

**Hörerkreis:**

Master

**Prüfungsform:**

Klausur

**nächster Wiederholungstermin:**

WS 2018/2019

**Modulzugehörigkeit:**

ESS 3.1 und ESS 3.3

**vorausgesetzte Kenntnisse:**

Rechnernetze

**Homepage:**

<https://cs.uni-paderborn.de/cn/teaching/courses/mobile-communication/>

## Software Analysis

**Dozent:** Wehrheim

**Büro:** O4.225

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

In the course, we will study techniques for (mostly) statically analysing properties of programs. Such information can be valuable for optimization phases of compilers (like knowledge about live variables or constant values of variables), but might also help verifying the correctness of programs, i.e., the adherence to certain safety requirements. The plan for the course is also to get to know and practically use different tools carrying out such software analysis.

### Verschiedenes

**Hörerkreis:**

Master Informatik

**Prüfungsform:**

Mündliche Prüfung

**Modulzugehörigkeit:**

Software Engineering

**vorausgesetzte Kenntnisse:**

Programmierung, Logik

# Kontextuelle Informatik

**Dozent:** Selke

**Büro:** F1.101

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Informatiker entwickeln auf Zeichen basierende Produkte (Programme, Spezifikationen, Dokumentationen etc.), die einen spezifischen Gegenstandsbereich modellieren. Bei der Entwicklung solcher Produkte stellen sich vielfältige Fragen: Wie können die zu verarbeitenden Daten sowie die umzusetzenden Prozesse angemessen modelliert werden? Welche Konsequenzen ergeben sich aus der Möglichkeit, Systeme interaktiv zu gestalten? Welche Rolle wird beim Einsatz der Software den Benutzern, welche der Software zuteil? Welche Rahmenbedingungen des Einsatzkontexts sind dabei zu beachten?

Die Veranstaltung »Kontextuelle Informatik« erörtert die im weiteren Verlauf relevanten Grundbegriffe der Informatik mit besonderem Augenmerk auf die Unterscheidung zwischen technischen Konzepten und der Nutzungssphäre. Vor diesem Hintergrund werden Theorien interaktiver Systeme betrachtet, um insbesondere die Rolle digitaler Medien für geistige Prozesse untersuchen. Bei der Entwicklung von Informatiksystemen müssen die relevanten Daten und Prozesse bis zu einem gewissen Grad antizipiert und als formales System beschrieben werden. Dies wirft Fragen auf, unter welchen Bedingungen eine solche formale Beschreibung adäquat erfolgen kann und welche Konsequenzen sich daraus in Bezug auf die Zuverlässigkeit und den verantwortbaren Einsatz von Informatiksystemen ergeben.

## Verschiedenes

**Hörerkreis:**

i-I, i-m, ie, ii, im, winf

**Prüfungsform:**

Modulprüfung

**vorausgesetzte Kenntnisse:**

Bachelor in Informatik oder vergleichbare Kenntnisse

**nächster Wiederholungstermin:**

WS 18/19

## Fundamentals of Model-Driven Engineering (in English)

**Dozent:** Anjorin

**Büro:** ZM1.03-06

**Sprechstunde:** Mo, 10-11 Uhr

### Inhaltsangabe

Model-Driven Engineering (MDE) focusses on the use of models, i.e., suitable abstractions, as primary artefacts in (software) engineering processes. Important tasks include specifying these models (metamodelling), deriving new models from other models (model transformation), and keeping different but related models consistent to each other (model synchronisation).

While it is important to develop an intuitive understanding for central concepts such as models, metamodels, and model transformations, it is equally important to establish solid formal foundations for at least the basic concepts. This is especially the case when providing reliable tool support (e.g., static analyses) for MDE activities.

This lecture, therefore, introduces basic MDE concepts including models, metamodels, and model transformations, providing a precise and detailed formalisation using very basic category theory (algebraic graph transformation).

The lecture is designed to be especially accessible to computer scientists, by providing a hands-on constructive mapping of all definitions and results to executable programs in a main stream OO language (Java). The model-driven development of a simple project management tool (<https://trello.com>) will be used as a „mini-project“ in the exercises to demonstrate and explore all concepts introduced in the lecture.

This lecture complements other MDSD lectures and can be taken before (providing foundations) or after (providing a specialization).

### Literaturangaben

- **Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.:** *Fundamentals of Algebraic Graph Transformation*, <http://www.springer.com/us/book/9783540311874>
- **D.E. Rydeheard, R.M. Burstall:** *Computational Category Theory*, <http://www.cs.man.ac.uk/~david/categories/book/book.ps>
- **Ehrig, H., Ermel, C., Golas, U., Hermann, F.:** *Graph and Model Transformation*, <http://www.springer.com/de/book/9783662479797>

### Verschiedenes

**Hörerkreis:**

Informatik Master, Mathematik Master

**Modulzugehörigkeit:**

Informatik Master, SWT (III.1.1, III.1.2, III.1.5, III.1.6)

**Prüfungsform:**

Mündliche Prüfung

**vorausgesetzte Kenntnisse:**

Gute Kenntnisse einer modernen objektorientierten Programmiersprache (zB. Java).

**nützliche Parallelveranstaltungen:**

Model-Driven Software Development (in English)

**weiterführende Veranstaltungen:**

Model-Driven Software Development (in English)

**nächster Wiederholungstermin:**

WS 18/19

**Homepage:**

[https://cs.uni-paderborn.de/dbis/  
lehre/veranstaltungen/ws-201718/  
fundamentals-of-model-driven-engineering/  
news/](https://cs.uni-paderborn.de/dbis/lehre/veranstaltungen/ws-201718/fundamentals-of-model-driven-engineering/news/)

## Advanced Software Engineering: Methods, Architectures, Industrial Applications (in English)

**Dozent:** Engels

**Büro:** ZM1.03-45

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

This course covers advanced and novel software engineering topics from a research as well as an industrial perspective. Lectures on basic concepts of development methods and architectures of software systems are paired with lectures on industrial applications and experience. These lectures on industrial applications are given by senior experts from industrial partners. They will illustrate the introduced concepts with several real life industrial examples.

The setting of this new course allows students to get insights into novel concepts of Software Engineering as well as into their usage in industrial contexts. It also allows students to come into direct contact with experts from their later working fields The course will cover topics like:

- Architecture (Service-oriented Architecture, Microservices, Web Services, Mobile Architectures, Architectural Frameworks)
- Methods (Agile Software Development Methods like Scrum, Situational Method Engineering, Requirements Engineering)
- Project Management (Effort Estimation, Economics of Software Projects)

### Literaturangaben

See website

### Verschiedenes

**Hörerkreis:**

Wirtschaftsinformatiker, Informatiker

**Prüfungsform:**

Oral Exam

**Homepage:**

<https://cs.uni-paderborn.de/dbis/lehre/veranstaltungen/ws-201718/advanced-software-engineering/>

**Modulzugehörigkeit:**

2009: SWT&IS Modul III.1.1, III.1.3, III.1.6

**vorausgesetzte Kenntnisse:**

Software Design, Database Systems, Modeling Languages (UML)

# Type Systems for Correctness and Security

**Dozent:** Hermann

**Büro:** ZM1.02-04

**Sprechstunde:** Mi, 15-16

## Inhaltsangabe

Type systems in programming languages prevent illicit behavior of programs from the very start. They provide valuable feedback for programmers to prevent bugs, crashes or even security vulnerabilities. In this lecture we will study the theory, the properties and the implementation of modern type systems. We will take a pragmatic view on type systems and will develop type checkers along our way in this course to put theory into practice immediately. We will also take a closer look at type systems of well-known programming languages like Java or Scala. We will cover the following topics (but are not limited to them):

- Static vs. dynamic type checking
- Operational semantics
- Soundness of type systems
- Type inference
- Polymorphism
- Subtyping
- Security and safety guarantees through type systems
- Path-dependent types
- Dependent types

## Literaturangaben

The main part of the course will roughly follow this book:

- **Pierce, Benjamin C.:** *Types and Programming Languages*, The MIT Press, 2002

Specific topics might be taken from this book:

- **Pierce, Benjamin C.:** *Advanced Topics in Types and Programming Languages*, The MIT Press, 2004

or from current/recent research.

## Verschiedenes

**Hörerkreis:**

Informatik Master

**Modulzugehörigkeit:**

III.1.3, III.1.2, III.1.5, III.1.6,

**Prüfungsform:**

mündliche Prüfung (nach PO)

**Leistungspunkte:**

6

**vorausgesetzte Kenntnisse:**

Aus der Vorlesung „Grundlagen der Programmiersprachen“: Spracheigenschaften, Syntaktische Strukturen, Datentypen, Funktionale Programmierung

From the lecture „Foundations of programming languages“: Language properties, Syntactical program structures, Data types and type hierarchies, Functional programming



## Seminar: Software Transactional Memory

**Dozent:** Wehrheim

**Büro:** O4.225

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

Software Transactional Memory (STM) provides programmer with a high-level synchronization technique for access to shared memory. In this seminar, we will look at STM algorithms as well as validation techniques (testing, verification) for showing correctness of STMs.

### Verschiedenes

**Hörerkreis:**

Master Computer Science

**Prüfungsform:**

Presentation + report + active participation  
+ reviews of other reports

**Vorbesprechung:**

to be announced, see webpages of research  
group

**Modulzugehörigkeit:**

Software Engineering

**vorausgesetzte Kenntnisse:**

courses Model Checking or Deductive Verifi-  
cation would be helpful, logic

# Seminar: Gems of Theoretical Computer Science

**Dozent:** Meyer auf der Heide

**Büro:** F1.301

**Sprechstunde:** Mi, 13-14 Uhr

## Inhaltsangabe

The seminar based on a set of selected papers and textbook sections that demonstrate the beauty of problem solutions in the field of Theoretical Computer Science. We will find out that the preoccupation with sophisticated proof techniques, elegant arguments and surprising constructions is highly enjoyable. The seminar is inspired by the book „Perlen der Theoretischen Informatik“ by Uwe Schöning, in which he presents a collection of results demonstrating highlights of Theoretical Computer Science. Of course, the selection of seminar topics is affected by the supervisor’s taste and research area. The seminar is expected to take place as a block seminar at the end of the lecture period.

## Literaturangaben

- **Uwe Schöning:** *Perlen der Theoretischen Informatik*, Spektrum Akademischer Verlag, 2002

## Verschiedenes

**Hörerkreis:**

Master Informatik

**Prüfungsform:**

seminar thesis, presentation

**nächster Wiederholungstermin:**

Winter Term 2018/19

**Homepage:**

<http://www.hni.upb.de/en/alg/teaching>

**Modulzugehörigkeit:**

Seminar: Modul III 2.1, 2.2, 2.3, 2.4

**vorausgesetzte Kenntnisse:**

Bachelor

**Vorbesprechung:**

first week of lectures, to be announced on lecture’s web site

## Seminar: Digital Humanities

**Dozent:** Selke

**Büro:** F1.101

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

Die Digital Humanities nutzen Methoden der Informatik und Techniken der digitalen Informationsverarbeitung, um kulturwissenschaftliche Forschungsprozesse zu unterstützen oder überhaupt erst möglich zu machen. Das beginnt z.B. mit der digitalisierten Archivierung und systematischen Speicherung von Texten oder Bildern, um Überlieferungen zu bewahren bis hin zur (automatisierten) Analyse und Auswertung dieser (zum Teil sehr großen) Datenbestände, um neue Informationen und Erkenntnisse daraus zu gewinnen. Teilnehmer des Seminars untersuchen Werkzeuge, Codierungsstandards, Infrastrukturarchitekturen und theoretische Konzepte aus dem Bereich der DH und bewerten diese aus dem Blickwinkel der Informatik. Das Seminar findet als Blockveranstaltung statt. Die Themenvergabe findet bei der Vorbesprechung statt.

## Seminar: Selected Topics in Wireless Networking

**Dozent:** Dressler

**Büro:** F1.401

**Sprechstunde:** nach Absprache

### Inhaltsangabe

This seminar is being offered as part of the “Master Seminar” series, each issue of which treats current topics in the context of active research domains.

### Verschiedenes

**Hörerkreis:**

Master Informatik, Master Computer Engineering

**nächster Wiederholungstermin:**

Summer 2018

**Vorbesprechung:**

erste Semesterwoche

**Homepage:**

<http://www.ccs-labs.org/teaching/ms/>

## PG Secure Integration of Cryptographic Software

**Dozent:** Bodden

**Büro:** ZM1.02-09

**Sprechstunde:** Nach Absprache

### Inhaltsangabe

Multiple studies have shown that Java Crypto APIs are often misused by application developers. Such misuses are caused by an interplay of inappropriate API Design and developers' lack of domain knowledge. These Crypto APIs are generally low-level, for example, they do not provide developers who want to include a file encryption in their application with a convenient method such as `encryptFile()`. Instead, the developers have to implement the encryption themselves by using cryptographic algorithms directly. Developers with only little or no experience in cryptography struggle with that level of abstraction. Often, they do not know which algorithms are secure under which circumstances and make easy mistakes, causing their implementation to be insecure.

Tool support for existing cryptographic APIs may help developers integrating cryptography securely into their application. We aim at lifting the level of abstraction to a more convenient level by generating task-based wrapper code for the APIs. Additionally, the tool may run static analyses on the code to alert the application developers when they are using an insecure algorithm or misuse the API.

To build such a tool, knowledge in variability modelling, code generation and analysis is required. In the first phase of the project group, you will familiarize yourself with said topics and prepare a first design of how to combine these techniques in order to provide effective tool support.

In the second phase, the group will split into several sub groups that work on smaller components of the tool. At the end of the year, all components must be integrated with each other to build a stable prototype.

### Verschiedenes

**Hörerkreis:**

Informatik Master, Elektrotechnik Master

**Modulzugehörigkeit:**

Projektgruppe (PG)

**Prüfungsform:**

Projektarbeit

**Leistungspunkte:**

15

**Homepage:**

<https://www.hni.uni-paderborn.de/swt/lehre/pg-sics/>

### Oberseminar: Algorithmen und Komplexität

**Dozent:** Meyer auf der Heide

**Büro:** F1.301

**Sprechstunde:** Mi, 13-14 Uhr

#### Inhaltsangabe

Im Oberseminar werden Arbeiten und Themen aus dem Bereich der Fachgruppe Friedhelm Meyer auf der Heide präsentiert. Mitarbeiter stellen aktuelle Ergebnisse und Fragestellungen ihrer laufenden Arbeiten vor. Studierende halten Abschlussvorträge ihrer Bachelorarbeiten und Masterarbeiten. Eingeladene Gäste präsentieren ihre Gastvorträge. Neue Originalarbeiten werden vorgestellt. Die Studierenden sind herzlich eingeladen an den Vorträgen teilzunehmen. Sie können sich durch das Seminar einen Überblick über aktuelle Forschungsfragen der Fachgruppen verschaffen. Momentane Schwerpunkte: Spieltheorie, Theorie paralleler Systeme, Randomisierte Algorithmen, Komplexitätstheorie, Drahtlose Netzwerke, Algorithmische Geometrie, Algorithmen in der Computergrafik.

#### Verschiedenes

**Hörerkreis:**

Bachelor, Master Informatik

**vorausgesetzte Kenntnisse:**

keine

**Homepage:**

[www.hni.upb.de/alg/oberseminar](http://www.hni.upb.de/alg/oberseminar)

**Prüfungsform:**

kein Schein

**nächster Wiederholungstermin:**

Seminar findet jedes Semester statt

## Oberseminar Informatik und Bildung

**Dozent:** Schulte

**Büro:** F2.116

**Sprechstunde:** Mo, 13-14

### Inhaltsangabe

Im Oberseminar werden aktuelle Bachelor-, Master- oder Studienarbeiten, Forschungsvorhaben und Projektberichte vorgestellt. Gelegentlich werden auch externe Gäste eingeladen, die Inhalte zu aktuellen Forschungsthemen vorstellen.

### Verschiedenes

**Homepage:**

<https://cs.uni-paderborn.de/ddi/lehre>

# Stufenbezogene Unterrichtsmodelle

**Dozent:** Schulte

**Büro:** F2.116

**Sprechstunde:** Mo, 13-14

## Inhaltsangabe

- Planung, Durchführung und Analyse von stufenbezogenen Unterrichtseinheiten (Sek I, Sek II, BK)
- Analyse und Reflexion von Lehr- und Lernprozessen
- Einsatz von Unterrichtssoftware und Lernumgebungen im Informatikunterricht (Sek I, Sek II, BK)
- Handlungsorientierter Informatikunterricht (z.B. Plan- und Rollenspiele, Informatik unplugged)
- Umgang mit Heterogenität (u. a. Leistungsdifferenzierung, Genderaspekte, Förderung)

## Verschiedenes

**Hörerkreis:**  
Lehramt Informatik

**Homepage:**  
<https://cs.uni-paderborn.de/ddi/lehre>



# Fachdidaktische Grundlagen

**Dozent:** Schulte

**Büro:** F2.116

**Sprechstunde:** Mo, 13-14

## Inhaltsangabe

### 1. Grundfragen der Informatikdidaktik

- Aufgaben und Methoden der Informatikdidaktik
- Informatikdidaktik und Fachwissenschaft Informatik
- Fachwissenschaftliches und Fachdidaktisches Selbstverständnis der Informatik
- Informatiksysteme aus didaktischer Perspektive

### 2. Informatikspezifische Medien und Methoden

- Sprachkonzepte im Informatikunterricht
- Medien im Informatikunterricht
- Formen der Unterrichtsorganisation im Informatikunterricht
- Technische Aspekte von Informatiksystemen im Informatikunterricht

### 3. Informatische Bildungskonzepte

- Informatikunterricht und Allgemeinbildung
- Informatikunterricht und Medienbildung
- GI-Gesamtkonzept informatischer Bildung / Bildungsstandards

### 4. Grundlagen der fachspezifischen Diagnostik im Informatikunterricht

- Informatikunterricht und Lerntheorien
- Leistungsdiagnostik im Informatikunterricht
- Evaluation von Informatikunterricht

## Verschiedenes

**Hörerkreis:**

Informatik Lehramt

**Homepage:**

<https://cs.uni-paderborn.de/ddi/lehre>

# Fachdidaktische Konzepte

**Dozent:** Schulte

**Büro:** F2.116

**Sprechstunde:** Mo, 13-14

## Inhaltsangabe

- Didaktische Konzeptionen  
hardwareorientierter Ansatz, algorithmenorientierter Ansatz, anwendungsbezogener / benutzerorientierter Ansatz, Konzept der fundamentalen Ideen, informationswissenschaftlicher Ansatz u.a., systemorientierte Didaktik der Informatik, theoretische Grundlagen des systemorientierten Ansatzes
- Methoden des Informatikunterrichts  
Modellierungstechniken im Informatikunterricht, unterrichtliche Phasenmodelle, Dekonstruktion in der Praxis des Informatikunterrichts
- Richtlinien / Rahmenlehrpläne  
internationale Diskussion (u.a. UNESCO / IFIP, ACM Curriculum), Informatik Lehrplan NRW (incl. Vorgaben zum Abitur), EPA Informatik, Lehrpläne anderer Bundesländer, Einfluss von Lehrplänen auf Unterrichtsmaterialien (Schulbücher, Software u.a.)
- Umgang mit Heterogenität
- Informatik Anfangsunterricht
- Geschlechtsspezifische Zugangsweisen

## Verschiedenes

**Hörerkreis:**  
Lehramt Informatik Master

**Homepage:**  
<https://cs.uni-paderborn.de/ddi/lehre>

## Grundlagen der Programmierung für Ingenieure II

**Dozent:** Fischer

**Büro:** F1.223

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

Die Veranstaltung richtet sich an Erstsemesterstudierende der Studienrichtungen Elektrotechnik. Das Hauptziel der Veranstaltung ist die Vermittlung von Grundkenntnissen der Programmiersprache C++. Dazu gehören einfache Anweisungen, Ablaufsteuerungen, Arrays, dynamische Datenstrukturen und objektorientierte Programmierung. Die Vermittlung dieser Grundkenntnisse wird von Übungen am Computer begleitet.

Für Studierende der Informatik, Wirtschaftsinformatik, Ingenieurinformatik usw., die die Veranstaltung im Rahmen des Studium Generale belegen möchten, wird keine Prüfungsmöglichkeit angeboten. Das gilt auch für Studierende der Mathematik. Mathematik Studierende seien auf die vergleichbare Veranstaltung der Mathematik „C-Programmierung Mathematik“ verwiesen. Für informatikferne Fächer wie Physik, Medienwissenschaften, Kulturwissenschaften usw. ist die Möglichkeit zur Prüfung im Rahmen Studium Generale gegeben. Im Zweifel kontaktieren Sie den Veranstalter.

### Literaturangaben

- **Ulrich Breymann:** *C++ Einführung und professionelle Programmierung*,
- **Bjarne Stroustrup:** *The C++ Programming Language*,
- **Bjarne Stroustrup, Petra Alm und Dirk Louis:** *Einführung in die Programmierung mit C++*,

### Verschiedenes

**Hörerkreis:**

1. Semester Maschinenbau, Wirtschaftsingenieurwesen, Elektrotechnik

**Modulzugehörigkeit:**

Grundstudium

**Prüfungsform:**

Klausur

**vorausgesetzte Kenntnisse:**

keine

**nächster Wiederholungstermin:**

WS 2018/19

**Homepage:**

<https://www.hni.upb.de/alg/lehre>

# Grundlagen der Programmierung für MB

**Dozent:** Fischer

**Büro:** F1.223

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Die Veranstaltung richtet sich an Erstsemesterstudierende der Studienrichtungen Maschinenbau, Wirtschaftsingenieurwesen und Elektrotechnik.

Das Hauptziel der Veranstaltung ist die Vermittlung von Grundkenntnissen einer Programmiersprache. Ab dem WS 12/13 ist das die Sprache C++. Dazu gehören einfache Anweisungen, Ablaufsteuerungen, Arrays, dynamische Datenstrukturen und objektorientierte Programmierung. Die Vermittlung dieser Grundkenntnisse wird von Übungen am Computer begleitet.

Für Studierende der Informatik, Wirtschaftsinformatik, Ingenieurinformatik usw., die die Veranstaltung im Rahmen des Studium Generale belegen möchten, wird keine Prüfungsmöglichkeit angeboten. Das gilt auch für Studierende der Mathematik. Mathematik Studierende seien auf die vergleichbare Veranstaltung der Mathematik „C-Programmierung Mathematik“ verwiesen. Für informatikferne Fächer wie Physik, Medienwissenschaften, Kulturwissenschaften usw. ist die Möglichkeit zur Prüfung im Rahmen Studium Generale gegeben. Im Zweifel kontaktieren Sie den Veranstalter.

## Literaturangaben

- **Ulrich Breymann:** *C++ Einführung und professionelle Programmierung*,
- **Bjarne Stroustrup:** *The C++ Programming Language*,
- **Bjarne Stroustrup, Petra Alm und Dirk Louis:** *Einführung in die Programmierung mit C++*,

## Verschiedenes

### Hörerkreis:

1. Semester Maschinenbau, Wirtschaftsingenieurwesen, Elektrotechnik

### Modulzugehörigkeit:

Grundstudium

### Prüfungsform:

Klausur

### vorausgesetzte Kenntnisse:

keine

### nächster Wiederholungstermin:

WS 2018/19

### Homepage:

<https://www.hni.upb.de/alg/lehre>

## Einführung in die Web-bezogenen Sprachen

**Dozent:** Feldmann

**Büro:** F2.401

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

Diese Vorlesung wird für Studierende des Studienganges Medienwissenschaften angeboten.

Ziele: Die Vorlesung soll Studierende dazu befähigen,

- Sprachen, die zur Entwicklung von Web-Präsenzen eingesetzt werden, zu verstehen, anzuwenden und zu beurteilen,
- einfache Web-Präsenzen mit den dafür heute gebräuchlichen Sprachen und Methoden zu entwickeln,
- Sprachen, die in Zukunft für solche Aufgaben eingesetzt werden, dann selbständig zu erlernen,
- grundlegende, allgemeine Programmiertechniken anzuwenden.

### Literaturangaben

- **G. Pomaska:** *Webseiten-Programmierung Sprachen, Werkzeuge, Entwicklung*, Springer Verlag, 2012
- **K. Tatroe, P. MacIntyre, R. Lerdorf:** *Programming PHP*, O'Reilly Verlag, 2013
- **R. Asleson, N.T. Schutta:** *Foundations of Ajax*, APress Verlag, 2006

### Verschiedenes

**Hörerkreis:**

MeWi

**Prüfungsform:**

Klausur

**vorausgesetzte Kenntnisse:**

Einführung in die Informatik für Geisteswissenschaftler

**nächster Wiederholungstermin:**

WS 14/15

**Homepage:**

in PAUL

# Datenverarbeitung

**Dozent:** Fischer

**Büro:** F1.223

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Die Veranstaltung richtet sich an Erstsemesterstudierende der Studienrichtungen Elektrotechnik und Wirtschaftsingenieurwesen.

Das Hauptziel der Veranstaltung ist die Vermittlung von Grundkenntnissen einer Programmiersprache C++. Dazu gehören einfache Anweisungen, Ablaufsteuerungen, Arrays, dynamische Datenstrukturen und objektorientierte Programmierung. Die Vermittlung dieser Grundkenntnisse wird von Übungen am Computer begleitet.

Für Studierende der Informatik, Wirtschaftsinformatik, Ingenieurinformatik usw., die die Veranstaltung im Rahmen des Studium Generale belegen möchten, wird keine Prüfungsmöglichkeit angeboten. Das gilt auch für Studierende der Mathematik. Mathematik Studierende seien auf die vergleichbare Veranstaltung der Mathematik „C-Programmierung Mathematik“ verwiesen. Für informatikferne Fächer wie Physik, Medienwissenschaften, Kulturwissenschaften usw. ist die Möglichkeit zur Prüfung im Rahmen Studium Generale gegeben. Im Zweifel kontaktieren Sie den Veranstalter.

## Literaturangaben

- **Ulrich Breymann:** *C++ Einführung und professionelle Programmierung*,
- **Bjarne Stroustrup:** *The C++ Programming Language*,
- **Bjarne Stroustrup, Petra Alm und Dirk Louis:** *Einführung in die Programmierung mit C++*,

## Verschiedenes

### Hörerkreis:

1. Semester Maschinenbau, Wirtschaftsingenieurwesen, Elektrotechnik

### Modulzugehörigkeit:

Grundstudium

### Prüfungsform:

Klausur

### nützliche Parallelveranstaltungen:

keine

### nächster Wiederholungstermin:

WS 2018/19

### Homepage:

<https://www.hni.upb.de/alg/lehre>

# Projektmanagement für Computer Engineering

**Dozent:** Karl

**Büro:** O3.158

**Sprechstunde:** Montag, 13 - 14 Uhr

## Inhaltsangabe

Grundlagen des Projektmanagements, insbes. IT-Projektmanagement. Die Veranstaltung „Projektmanagement“ vermittelt die theoretischen Grundlagen für das Management von Entwicklungsprojekten im IT-Bereich.

Im Einzelnen werden folgende Themen behandelt:

- Projektmanagement für eine Person: Zeitplanung, Deadlines, Milestones, Budget und Ressourcen, Abhängigkeiten und kritischer Pfad.
- Projektmanagement im Team: Teamorganisation, Rollen, Motivation und Gesprächsführung.
- Werkzeuge und methodische Ansätze: Gantt- und Pert-Diagramme, Reporting, SCRUM, Agile-Development, Extreme Programming, etc.

## Verschiedenes

**Hörerkreis:**

Bachelor

**nächster Wiederholungstermin:**

WS 2018/2019

**Modulzugehörigkeit:**

Software- und Systementwurf

**Homepage:**

<https://cs.uni-paderborn.de/cn/teaching/courses/projektmanagement/>

# Evaluation des Workloads von Lerngruppen

**Dozent:** Gerstenberger

**Büro:** F2.201

**Sprechstunde:** nach Vereinbarung

## Inhaltsangabe

Die Veranstaltung dient der anonymen Erfassung von Zeitbudgets und Ergebnissen aus Übungszetteln und Klausuren von unbetreuten Lerngruppen als Vergleich zu den Daten aus der Veranstaltung „Agile Organisation (des Studienalltags) von Lerngruppen“.

Im Rahmen dieser Veranstaltung sollen der Arbeitseinsatz und Studienerfolg von Lerngruppen evaluiert werden. Voraussetzung dafür wäre die Teilnahme jeweils vollständiger Lerngruppen. Bei der Anmeldung zu dieser Veranstaltung wird versucht dies sicherzustellen, bzw. es könnte auch eine Option sein, im Rahmen der Anmeldung Lerngruppen zu bilden.

Inhalte:

- Evaluation
  - Anonyme Erfassung von Zeitbudgets
  - Anonyme Erfassung von Ergebnissen der Übungszettel und der Klausuren

Studienleistung:

Aktive Teilnahme an der semesterbegleitenden Evaluation.

## Verschiedenes

**Hörerkreis:**

B.Sc. Informatik, B: Sc. Computer Engineering, B.Ed. HRSGe, B.Ed. BK, B.Ed. Gy/Ge

**Prüfungsform:**

Studienleistung

**Leistungspunkte:**

1 ECTS im Studium Generale

**vorausgesetzte Kenntnisse:**

keine



## Agile Organisation von Lerngruppen

**Dozent:** Gerstenberger

**Büro:** F2.201

**Sprechstunde:** nach Vereinbarung

### Inhaltsangabe

Zusammen mit den Studierenden wird eine agile Planungsweise (angelehnt an Scrum) erarbeitet, mit der der Studienalltag organisiert werden soll. Die studentische Lerngruppe übernimmt dabei die Rolle des Scrum Teams, das interdisziplinär und selbstorganisiert seine Arbeiten plant und umsetzt. Während der Umsetzung werden die Zeitbudgets und die Ergebnisse aus den Übungszetteln und Klausuren anonym erfasst und evaluiert.

Im Rahmen dieser Veranstaltung sollen der Arbeitseinsatz und Studienerfolg der Lerngruppen evaluiert werden. Voraussetzung dafür wäre die Teilnahme jeweils vollständiger Lerngruppen. Bei der Anmeldung zu dieser Veranstaltung wird versucht dies sicherzustellen, bzw. es könnte auch eine Option sein, im Rahmen der Anmeldung Lerngruppen zu bilden.

Fachliche Kompetenzen:

Dieses Modul soll die Teilnehmer befähigen,

- Tages-, Wochen und Semester-Ziele zur erarbeiten und reflektieren
- Wochenpläne zu erstellen, diskutieren und reflektieren
- Tagespläne zu detaillieren, einschätzen, umzusetzen und anschließend zu reflektieren

Spezifische Schlüsselkompetenzen:

- Kooperations- und Teamfähigkeit entwickeln
- Lernstrategien kennenlernen, einsetzen und reflektieren
- Eigene Arbeitsverhalten einschätzen, reflektieren und optimieren

Inhalte:

- Empirische Weiterentwicklung einer agilen Vorgehensweise zur Organisation des Studienalltags
- Umsetzung der Vorgehensweise in Lerngruppen
- Reflektieren der Vorgehensweise (an sich und auf einer Metaebene)
- Evaluation
  - Anonyme Erfassung von Zeitbudgets
  - Anonyme Erfassung von Ergebnissen der Übungszettel und der Klausuren

Studienleistung:

Aktive Teilnahme und Gestaltung der Sprint Planning-, Daily Standup- und Sprint Review-Meetings. Teilnahme an der Gruppenarbeit und der semesterbegleitenden Evaluation.

### Verschiedenes

**Hörerkreis:**

B.Sc. Informatik, B: Sc. Computer Engineering, B.Ed. HRSGe, B.Ed. BK, B.Ed. Gy/Ge

**Prüfungsform:**

Studienleistung

**Leistungspunkte:**

4 ECTS im Studium Generale

**vorausgesetzte Kenntnisse:**

keine

**Vorbesprechung:**

nach Vereinbarung

## 5 Raum für Notizen

# Stundenplan

Uhrzeit	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
7 - 8					
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 13					
13 - 14					
14 - 15					
15 - 16					
16 - 17					
17 - 18					
18 - 19					
19 - 20					