Haskell

Bäume, mapTree, reduceTree / foldTree, treeSum, treeConcat, treeToList	WS 11 / 12 (Nr. 1)
Hirsch-Index, Bäume, allTrees, trees_4_13	SS 12 (Nr. 1)
Lauflängenenkodierung, splitWhen, group, encode	WS 12 / 13 (Nr. 1)
Java-Bytecode Erzeugung	WS 12 / 13 (Nr. 8)
Laziness, Streams, pythagoreisches Tripel	SS 13 (Nr. 1)
splits, alle möglichen Teillisten	SS 13 (Nr. 4)
Hamming-Zahlen	WS 13 / 14 (Nr. 1)
Kombinatoren, unendliche Liste, Eulerverfahren, iterate, zipWith	WS 13 / 14 (Nr. 2)
last (Listen-Element), inits	SS 14 (Nr. 1)
lexikalische Analyse, Maybe, Automat, Zahlensystem	SS 14 (Nr. 2)
Newton-Iteration	WS 14 / 15 (Nr. 1)
binäre Bäume, deleteMin, merge Trees	WS 14 / 15 (Nr. 2)
splitWords (Word, NonWord), joinWords, capitalize	SS 15 (Nr. 1)
Potenzreihe, iterate, addieren, multiplizieren, differenzieren, evaluieren	WS 15 / 16 (Nr. 1)
Datentyp, Typ-Instanziierung	WS 15 / 16 (Nr. 2)
Mehrwegbäume, treeIndex, treePositions, changeTree , overrideTree, i-tes Listen-Element	SS 16 (Nr. 1)
semimagische Quadrate, Duplikate-Test, transponieren, Überprüfung auf semimagisches Quadrat	SS 17 (Nr. 1)
Mehrweg-Bäume, tote & lebendige Blätter, prune	SS 17 (Nr. 2)

Prolog

Nichtdeterminismus, Wechselgeld, removeMoney, changeMoney	WS 11 / 12 (Nr. 2)
Wolf, Ziege, Kohl	SS 12 (Nr. 2)
Labyrinth	WS 12 / 13 (Nr. 3)
splits, alle möglichen Teillisten	SS 13 (Nr. 4)
Differenzlisten	SS 13 (Nr. 5)
reguläre Ausdrücke	WS 13 / 14 (Nr. 5)
Java-Bytecodeerzeugung	SS 14 (Nr. 5)
del, delete, freie Variablen in Lambda-Ausdruck	WS 14 / 15 (Nr. 4)
Haus vom Nikolaus, Eulerpfade, swap (Kanten umdrehen)	SS 15 (Nr. 4)
Rucksack-Problem, alle möglichen Teil-Listen	WS 15 / 16 (Nr. 3)
nichtdeterministische endliche Automaten	SS 16 (Nr. 2)
Noten-Datenbank, Campus-Verwaltungssystem, transitive Voraussetzung	SS 16 (Nr. 3)
freie Variablen	SS 17 (Nr. 3)

λ-Kalkül

Fixpunktkombinator, β-Reduktion	WS 11 / 12 (Nr. 3)
Kombinatoren, β-Reduktion, S K I	SS 12 (Nr. 3)
Church-Paare, fst, snd, map, allgemeinster Typ	WS 12 / 13 (Nr. 2)
Church-Paare, next, pred	SS 13 (Nr. 2)
Church-Zahlen, isZero, pred, add	WS 13 / 14 (Nr. 3)
Scott-Zahlen, isZero, pred, sub	SS 14 (Nr. 3)
freie Variablen, Redex, call-by-name, Normalform, α- & η-Äquivalenz	WS 14 / 15 (Nr. 3)
S K I, Normalform	SS 15 (Nr. 2)
Pair, fst, snd, swap (fst & snd v. pair), Normalform, euklidischer Algo, größter gemeinsamer Teiler, Church-Zahlen	WS 15 / 16 (Nr. 4)
Church-Zahlen, inc, add, get, react	SS 16 (Nr. 4)
Church-Zahlen, c_{true}, c_{false}, pred, Gleichheits-Test	SS 17 (Nr. 6)

Typinferenz

S K I, Let-Polymorphismus	WS 11 / 12 (Nr. 4)
Unifikation, Prolog-Notation	SS 12 (Nr. 4)
allgemeinster Typ, Typschema, let-Polymorphismus	SS 13 (Nr. 3)
let-Polymorphismus, λ-gebunde Variable $\rightarrow$ nicht polymorph	WS 13 / 14 (Nr. 4)
Scott-Zahlen, allgemeinster Typ, mgu	SS 14 (Nr. 4)
mgu, τ_{just}, let-Polymorphismus	WS 14 / 15 (Nr. 5)
mgu, let-Polymorphismus, λ-gebunde Variable $\rightarrow$ nicht polymorph	SS 15 (Nr. 3)
mgu, allgemeinster Typ, Typabstraktion, let-Polymorphismus	WS 15 / 16 (Nr. 5)
C_0 & C_{let}, let-Polymorphismus, mgu	SS 16 (Nr. 5)
Unifikation, ausgeschrieben	SS 17 (Nr. 4)
λ-Term zu Typ erstellen, let-Polymorphismus, allgemeinster Typ	SS 17 (Nr. 5)

Parallel Basics

Flynn's Taxonomie	WS 13 / 14 (Nr. 9) SS 14 (Nr. 6)
Vorteile, Risiken, Thread-Scheduling, Sperren, Race-Bedingung, Deadlock, blocking vs. non-blocking, synchron-Send	WS 14 / 15 (Nr. 6)
Beschleunigung berechnen, Amdahls Law	WS 14 / 15 (Nr. 7)
Synchronisierung, Bank, Race-Condition, Deadlock, synchronized, Coffman-Bedingungen	WS 15 / 16 (Nr. 7)

C

- Zeiger-Arithmetik, Arrays — WS 11 / 12 (Nr. 5), SS 12 (Nr. 5)
- Precedence-Rule — WS 12 / 13 (Nr. 4), SS 13 (Nr. 6), WS 13 / 14 (Nr. 6), SS 14 (Nr. 9)

MPI

Array-Durchschnitt	WS 11 / 12 (Nr. 6)
Matrix-Multiplikation	SS 12 (Nr. 6), SS 14 (Nr. 8)
Broadcast-Implementierung	WS 12 / 13 (Nr. 6)
Reduce, Reduce-Implementierung	SS 13 (Nr. 7)
Allgather & Alltoall	WS 13 / 14 (Nr. 7)
Scatterv / Gatherv	SS 14 (Nr. 8)
Send, Recv, Finalize, Parameter-Beschreibungen	WS 14 / 15 (Nr. 9)
Gather-Implementierung	SS 15 (Nr. 5)
Scatter, Allgather	WS 15 / 16 (Nr. 6)
Scatter, Reduce, Lückentext	SS 16 (Nr. 8)
Bcast, Reduce	SS 17 (Nr. 7)

Java

Synchronisierung, Bank, Race-Condition, Deadlock, synchronized, Coffman-Bedingungen	WS 15 / 16 (Nr. 7)
Zähl-Semaphore, Monitore, Signalisierungsmethoden	WS 15 / 16 (Nr. 8)
puffernder asynchroner Schreiber, flushBuffer, isInterrupted	SS 16 (Nr. 6)
ArrayList, synchronized, Deadlock, Race-Condition	SS 16 (Nr. 7)
Worker-Balancer, Amdahls Law	SS 17 (Nr. 8)
Petri-Netze, Token, Place, Deadlock, Coffman-Bedingungen	SS 17 (Nr. 9)

Design by Contract

Optionen selektieren, deselektieren, if-else: SS 17 (Nr. 10)

Compiler

Syntaktische Analyse, abstrakte Syntax, rekursiver Abstiegsparser	WS 11 / 12 (Nr. 8) SS 12 (Nr. 6)
Haskell, Java-Bytecode Erzeugung	WS 12 / 13 (Nr. 8)
First-& Follow-Mengen	WS 12 / 13 (Nr. 9) SS 14 (Nr. 10) SS 17 (Nr. 11)
Linksfaktorisierung, Haskell-Datentyp, Abstiegsparser	SS 13 (Nr. 9)
Abstiegsparser	WS 13 / 14 (Nr. 10)
Linksfaktorisierung, SLL(1), abstrakte Syntax, Klassenherarchie, Abstiegsparser	WS 14 / 15 (Nr. 10)
Linksfaktorisierung, abstrakte Syntax, Abstiegsparser, abstrakter Syntaxbaum	SS 15 (Nr. 2)
Abstiegsparser	WS 15 / 16 (Nr. 10)
Java-Bytecode, if	SS 16 (Nr. 9)