COMP4161
Advanced Verification (25)
lambda calculus
natural deduction for propositional logic
quantifiers, automation
confluence, congruence
inductive predicates, primitive recursion
regular expressions
semantics

COMP1911
Computing 1A (350+120)
compilers,makefiles,debugging
C,if,else,loops,functions,recursion
chars, arrays, 2d arrays
memory, addreses, pointers
strings, args

COMP1917
Computing 1 (280+100)
compilers,makefiles,debugging
C,if,else,loops,functions,recursion
chars, arrays, 2d arrays
memory, addresses, pointers
strings, args
structs
linked lists, BST
binary, hex, 2's-complement
stack frames, machine language

COMP1921
Computing 1B (50+130)
structs
linked lists, BST
complexity, Big-O, ADTs
AVL-trees, 234-trees??
Graphs,BFS,DFS,DAG's,MST??

COMP1927
Computing 2 (70+170)
deep understanding of C programming
complexity, Big-O, ADTs
AVL-trees, 234-trees
Graphs,BFS,DFS,DAG's,MST
heaps, sorting, recursion
Hash tables, skip lists

COMP3153
Algorithmic Verification (10)
automata and temporal logic
CTL and LTL model checking
Uppaal and fairness
static analysis
Syntactic Software Model Checking
SAT solving, SAT-based model checking

COMP3151
Foundations of Concurrency (20)
interleaving, interference freedom
temporal logic, fairness
invariants, progress properties
Szymanski's algorithm
semaphores
monitors, message passing
compositionality, termination
commitment and consensus
distributed algorithms

COMP3171
Object-Oriented Programming (110)
iterator, sequential containers, associative containers, adaptors
STL algorithms, function and class templates
copy control, overloaded operators, conversions
inheritance, dynamic binding, object model
memory management, exception handling
metaprogramming, multiple inhertiance
project work: C++

COMP2911
Engineering Design in Computing (120+70)
Java
object-orientation, classes, inheritance
OO design, interfaces, patterns
GUI, divide & conquer
search - DFS,BFS,IDS

COMP3161
Concepts of Programming Languages (40)
syntax, operational/axiomatic/denotational semantics
paradigms: imperative, OO, functional, concurrent, parallel
dynamic/strong typing, polymorphism, overloading
automatic memory management, multicore parallelism
project work: Haskell (no prior experience needed)

COMP4181
Language-based Software Safety (15)
advanced type systems
domain-specific languages
program transformations
purity and controlled effects
parallelism and concurrency
project work: Haskell (no prior experience needed)

COMP3152
Comparative Concurrency Semantics (5)
process algebra
semantic models of concurrent systems
semantic equivalences
modal and temporal logic
algorithmic verification
reasoning about knowledge

COMP4001
Object-Oriented Software Development
concurrent object models, threads
immutable objects, functions as objects
project work: Scala

COMP3131
Programming Languages & Compilers (50)
finite state automata, regular expressions
formal language specification strategies
context-free grammars, top-down vs bottom-up parsing
attribute grammars, type checking,
machine-level languages, JVM, visitor design pattern
project work: java