Functional Programming (FP) and Logic Programming (LP) permit values (FP) and variable bindings (LP) to be computed from a mathematical-logical, yet executable, specification. These paradigms allow easier analysis, verification, transformation, and maintenance of programs. Principles of FP, LP, and their combination to Functional-Logic Programming (FLP) will be visualized with data flow diagrams, color-'moded' variables, and animated computation traces. For example, in FLP, functions like the noted 'serialise' -- even when mapping from ground (variablefree) lists to ground lists -- can be more easily defined using intermediate non-ground lists. These are 'partial data structures' which contain logic variables as placeholders that can be instantiated by any later computation steps. FLP slides (http://www.cs.unb.ca/~boley/FLP/cs6715FLP.ppt) with animations and the online interpreter of our FLP implementation Relfun (http://www.dfki.uni-kl.de/~vega/relfun-cgi/cgi-bin/rfi.cgi) are available.

Wednesday March 17th @ 3:30pm
Information Technology Center, C-317