This tutorial/seminar will review the field of automated program synthesis over the last fifteen years with an emphasis on research systems developed at Kestrel Institute, particularly KIDS and Specware.

Program synthesis is the construction of code that correctly and efficiently implements a formal specification of input/output behavior. There are several stages that comprise the synthesis process. First, the construction of a correct program requires inference within a domain theory which formally models the application domain. Developing such a domain theory is a major and irreducible task in software engineering. Formal specifications of the software’ intended behavior is then written in the (domain-specific) vocabulary of the domain theory. Next, the specification is transformed to source-level code via a sequence of correctness-preserving transformations or refinements. These transformations/refinements embody knowledge about software architectures, algorithms, code optimization techniques, data structures, parallelism, etc. Finally the software is compiled to the target machine. The systematic application of representations of programming knowledge can allow the synthesis of complex codes embodying algorithms, data structures, and code-optimization techniques that might be too difficult to produce manually.

Various approaches to mechanized software development have been explored, including (1) deductive synthesis which focuses on the extraction of correct programs from a proof that the formal specification is consistent; (2) program transformation approaches that use syntactically-oriented rules incrementally to rewrite the specification into code; and (3) refinement approaches, that seek to refine one specification to another whose models are a subset of the former, ultimately refining to a single model of the initial specification.

KIDS (Kestrel Interactive Development System) builds on the strengths of all three approaches, in that it uses deductive tools to support algorithm design and optimization, transformation rules to optimize high-level designs, and tactics to automate complex refinement steps. KIDS has been used to derive over 70 algorithms for a wide variety of application domains, including scheduling, combinatorial design, sorting and searching, computational geometry, pattern matching, and mathematical programming. In recent years we have used KIDS to develop a variety of high-performance schedulers for various
end-users, including transportation scheduling and the scheduling of outage periods at power plants. By applying best-practice algorithm knowledge to scheduling problems, we have generated codes that run orders of magnitude faster than currently deployed systems.

Specware is a new system at Kestrel that builds on lessons learned from KIDS and other Kestrel research systems. Specware is based on concepts from mathematical logic, algebraic specifications, category and sheaf theory. It supports the composition of higher-order algebraic specification via co-limits, the refinement of specifications via specification morphisms, and the generation of code via institution morphisms.