

**MODULE:** **FORMAL SPECIFICATIONS**

**CODE:** **BSCH-4-2-10**

**Stage:** IV

**Credit Points:** 4 semester credits / 6 quarter units

### Overview and Aims

This module aims to: build on the work covered in Formal Design Methods by introducing students to formal methods for specifying software systems; teach the mathematics necessary to write software specifications; introduce a formal notation for system specification; use a specification tool kit to write specifications and use a theorem prover to prove them correct; develop case studies; show you how to prove properties of specifications formally; derive programs from specifications; explain design by contract; explain how design by contract can be used as a possible strategy for the implementation of specifications.

Upon successful completion of this module, you should be able to:

1. explain the features of state-based specification notations
2. write formal specifications of software systems using a mathematical modelling notation
3. use tools to prove specifications correct
4. prove properties of specifications using the mathematics covered in the module
5. use data refinement to derive implementable structures from mathematical constructs
6. refine a given specification through a sequence of steps to an implementation

### Module Content

### **Mathematical reasoning**

Propositional and Predicate Calculus; rules of inference; assertions over sequences; trading laws; quantifiers  $\forall, \exists, +, *$ , max, min - reasoning with quantifiers; Set notation; axioms and set operators, set theorems; theory of bags; theory of sequences; binary relations; relations: domain restriction, range restriction, domain subtraction, range subtraction, relational image, relational inverse, composition of relations; functions, partial functions, total functions, injective, surjective, bijective function; writing simple specifications with sets, relations, functions and sequences; using definitions to prove simple properties of specifications.

### **The *Perfect Developer* specification notation**

Properties of state based specification notations. Examples of state based specification notations.

The *Perfect Developer* notation: abstract machines, parameterized machines; states, constraints on states; data types; data structures - sets, relations, functions, arrays; statements - skip, if..else, case .., invariants, choice, select, pre, post, sees, uses, includes; multiple inclusion. Case studies. Proof obligations. Data refinement. Design by Contract. Implementation of design by contract in *Perfect Developer*. Implementing specifications with design by contract.