As an example, we can use specification by contract to define the getCharges() operation (in the Bike class) which works out the cost of hiring a bike for a given number of days.

- getCharges(no.Days) : (deposit, dailyHireRate, total)
- This operation works out the cost of hiring a particular bike for a given number of days
- The bike details must have been found and the requested number of days of hire known
- The Bike object attribute dailyHireRate is multiplied by the number of days (no.Days). The result is added to the deposit to give the total. The operation returns the deposit, the dailyHireRate and the total
- This operation does not call any others
- This operation does not change the values of any attributes.

The internal logic of an operation can be described in a number of ways depending on the complexity of the algorithm involved. One of the most popular approaches is to use semi-formal, structured English. Structured English is a limited and structured subset of natural language, with a syntax that is similar to that of a block-structured programming language. Structured English generally includes the following constructs:

- A sequence construct:
  e.g. the second statement below is executed immediately after the first statement;
  Get bikeDetails[3]
  Get hireCharges
- Two decision constructs:
  e.g. IF customer is existing customer
  THEN confirm customerDetails
  ELSE record customerDetails
  or: CASE customer is existing customer, confirm customer details
- Two repetition constructs:
  e.g. WHILE more bikes to add DO enter bikeDetails
  or: REPEAT enter bikeDetails UNTIL no more bikes to add
- comments enclosed in parentheses;
  (* this is a comment *)

3. *Nouns that are in the data dictionary are written as they appear there.*