

```

/* http://saravani.wordpress.com */

enum BinaryOperation {
    ADDITION = 1,
    SUBTRACTION = 2,
    MULTIPLICATION = 3,
    DIVISION = 4,
    MODULUS = 5
}

/** Structs are the basic complex data structures. They are comprised of fields
 * which each have an integer identifier, a type, a symbolic name, and an
 * optional default value. */

struct ArithmeticOperation {
    1:BinaryOperation op,
    2:double lh_term,
    3:double rh_term,
}

/* Structs can also be exceptions, if they are nasty. */
exception ArithmeticException {
    1:string msg,
    2:optional double x,
}

service Calculator {

    /**
     * A method definition looks like C code. It has a return type, arguments,
     * and optionally a list of exceptions that it may throw. Note that argument
     * lists and exception lists are specified using the exact same syntax as
     * field lists in struct or exception definitions.
     */

    double calc(1:ArithmeticOperation op) throws (1:ArithmeticException ae),

    /**
     * This method has an oneway modifier. That means the client only makes
     * a request and does not listen for any response at all. Oneway methods
     * must be void.
     *
     * The server may execute async invocations of the same client in parallel/
     * out of order.
     */

    oneway void run_task()
}

/* http://saravani.wordpress.com */

```