The abstract syntax for programs to be interpreted is given by the following datatypes:

```
(define-datatype program program?
  (a-program
    (exp expression?)))

(define-datatype expression expression?
  (lit-exp
    (datum number?))
  (var-exp
    (id symbol?))
  (primapp-exp
    (prim primitive?)
    (rands (list-of expression?))))
```

We evaluate a program by evaluating the expression in its body relative to the initial environment:

```
(define eval-program
  (lambda (pgm)
    (cases program pgm
      (a-program (body)
        (eval-expression body (init-env))))))
```

The main part of the interpreter is about expression evaluation:

```
(define eval-expression
  (lambda (exp env)
    (cases expression exp
      (lit-exp (datum) datum)
      (var-exp (id) (apply-env env id))
      (primapp-exp (prim rands)
        (let ((args (eval-rands rands env)))
          (apply-primitive prim args))))))
```

To evaluate a list of operands, we map `eval-expression` with a fixed environment over it:

```
(define eval-rands
  (lambda (rands env)
    (map (lambda (x) (eval-expression x env)) rands)))
```

The following deals with the application of primitive operators:

```
(define apply-primitive
  (lambda (prim args)
    (cases primitive prim
      (add-prim  () (+ (car args) (cadr args)))
      (subtract-prim () (- (car args) (cadr args)))
      (mult-prim  () (* (car args) (cadr args)))
      (incr-prim  () (+ (car args) 1)))))
```

Using DrScheme: To run DrScheme on the School's Unix system, type "setup DrScheme" to set it up and "drscheme" to run it. To use DrScheme on a private machine, download it from

www.cs.rice.edu/CS/PLT/packages/drscheme/download.html

Running the code in DrScheme: Some required constructs like "define-datatype" are not part of the standard definition of Scheme. They are available from the web-site "www.cs.indiana.edu/eopl/" of the book "Essentials of Programming Languages" on which this course is based. Follow the link called "directory" on that website, download the files "pltscheme-init.scm", "define-datatype.scm", "sllgen.scm", "test-harness.scm", "test-suite.scm", and "2-3-2.scm", and place them into the same directory as your code. At the beginning of your code, insert the following lines:

```
(load "pltscheme-init.scm")
(load "2-3-2.scm")
```

Before you let the code execute, select

Language → Choose Language → Full Scheme

from the menu of DrScheme. In the Windows version, debugging must be switched off due to some bug.