MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.090—Building Programming Experience
IAP 2006

Problem Set 1

Problems

Do these problems in the interaction window of DrScheme. You should open all a another fle called "hw1.scm". Copy the answers for the following problems into this file. file. Your should turn in a print-out of this file and your answer to problem 2, which will be on a separate sheet of paper.

1. *Evaluation* - For each expression:
   (a) Write the type of the expression.
   (b) Write your guess as to the expression's return value. If the expression is erroneous indicate "error" for the value and include a brief explanation why. If the expression returns an unspecified value, write whatever you want!
   (c) Evaluate the expression, and copy the response from the interaction window.

```
4

5.5

4.2e1

(+ 1 2)

(7)

(* (+ 7 8) (   -  5 6))
```

2. (Do this problem on a separate sheet of paper) Assume the following expressions have been evaluated:

```
(define red 44)

(define green 43)

(define blue green)

(define purple (+ blue green))
```

   Write a table for the names and values that were created by evaluating these expressions.

   Now write the table after the following expressions are evaluated

```
(define op *)
(define * 3)
```

What will this expression evaluate to?

```
(op * green)
```

3. - write an expression that evaluates to 3.

4. - write a *more interesting* expression that evaluates to 3.

5. *Define X* - for each of the following expressions:
   (a) Identify the variables that are *unbound*.
   (b) Supply definitions (ie `(define x ...)`) for each of the variables that make the expression evaluate to the target value.
   (c) Type in the expressions and verify that your solution gives the correct result.

```
(+ x (* y 3))
;Value: 13
```

```
(= yum (* -1 (+ yum 2)))
;Value: #t
```

```
(* cm-per-inch inch-per-foot)
30.48
```

6. *Primitive Procedures* - for each of the following expressions:

   (a) Identify the primitive procedures which you don't already know

   (b) Write down a guess as to what the primitive procedure does.

   (c) Look it up in the MITScheme reference manual (you may find the index handy).

   (d) Write an example usage of the procedure and test it to see that it works as you suspect it does.

   (e) Fill in the blanks in the original expression such that it evaluates to the target value.

   Reminder: Primitive procedures are operations that are built into Scheme.

```
(+ 3 (abs ________))
;Value: 5


(string-append "foo" ___ "baz")
;Value: "foobarbaz"

(string-append
  (number->string
    (gcd 35 _____))
  " rules!")
;Value: "7 rules!"

(define s _______)
(+ 7 (string-search-forward "ow" s))
;Value: 14
```