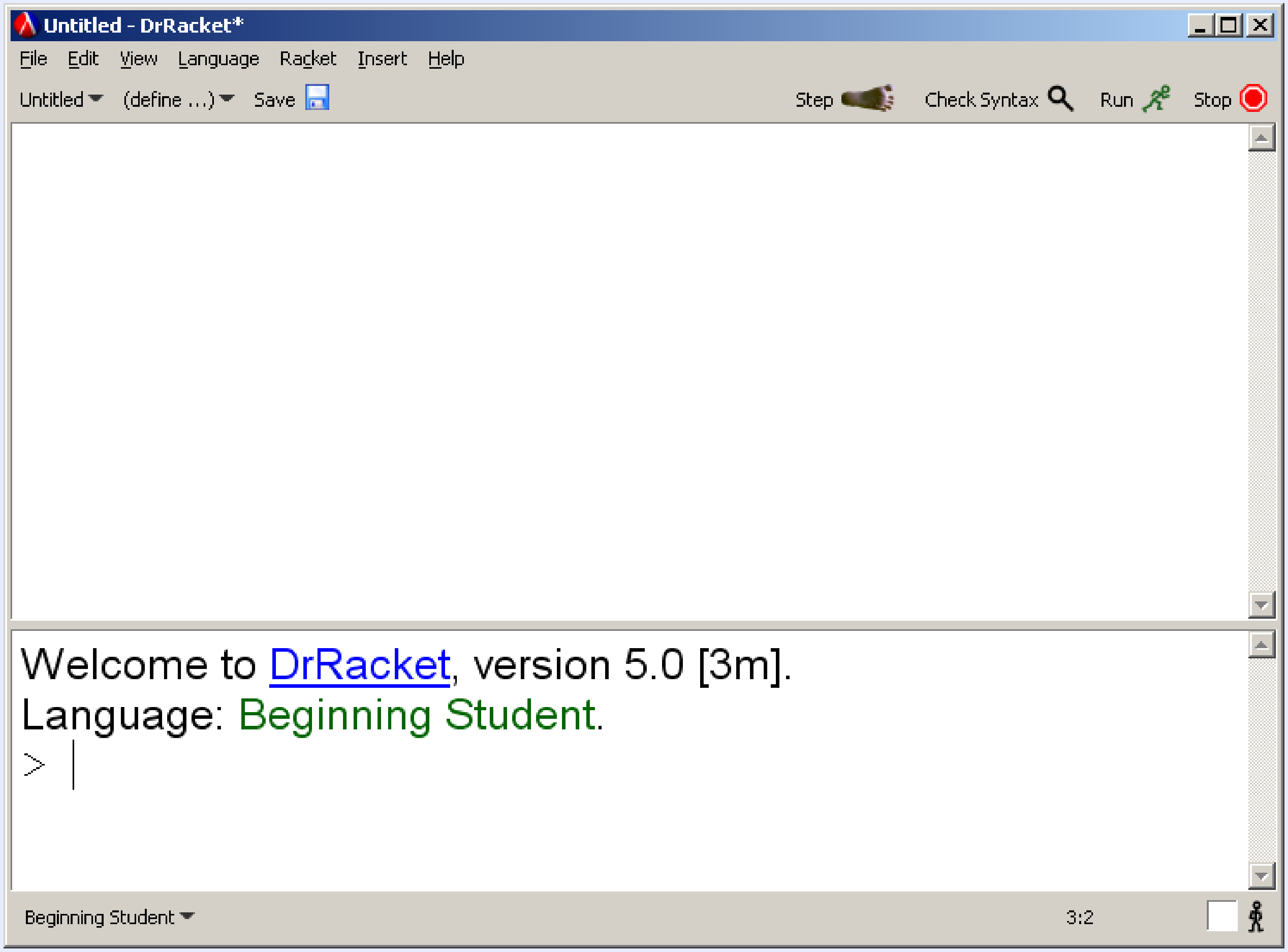
On The Design of Error Messages Aimed at Novice Programmers

Guillaume Marceau, Kathi Fisler (WPI)
Shriram Krishnamurthi (Brown)

Workshop on Scheme and Functional Programming
Montreal, 2010



```
Welcome to DrRacket, version 5.0 [3m].
Language: Beginning Student.
>
```

Beginner Student Language

```
(define (add-numbers)
  (5 + 3))
```

define: expected at least one argument name after the function name, but found none

```
(define (add-numbers x y)
  (x + y))
```

function call: expected a defined name or a primitive operation name after an open parenthesis, but found a function argument name

Advanced Student Language

```
(define (add-numbers)
  (5 + 3))
```

OK

```
(define (add-numbers x y)
  (x + y))
```

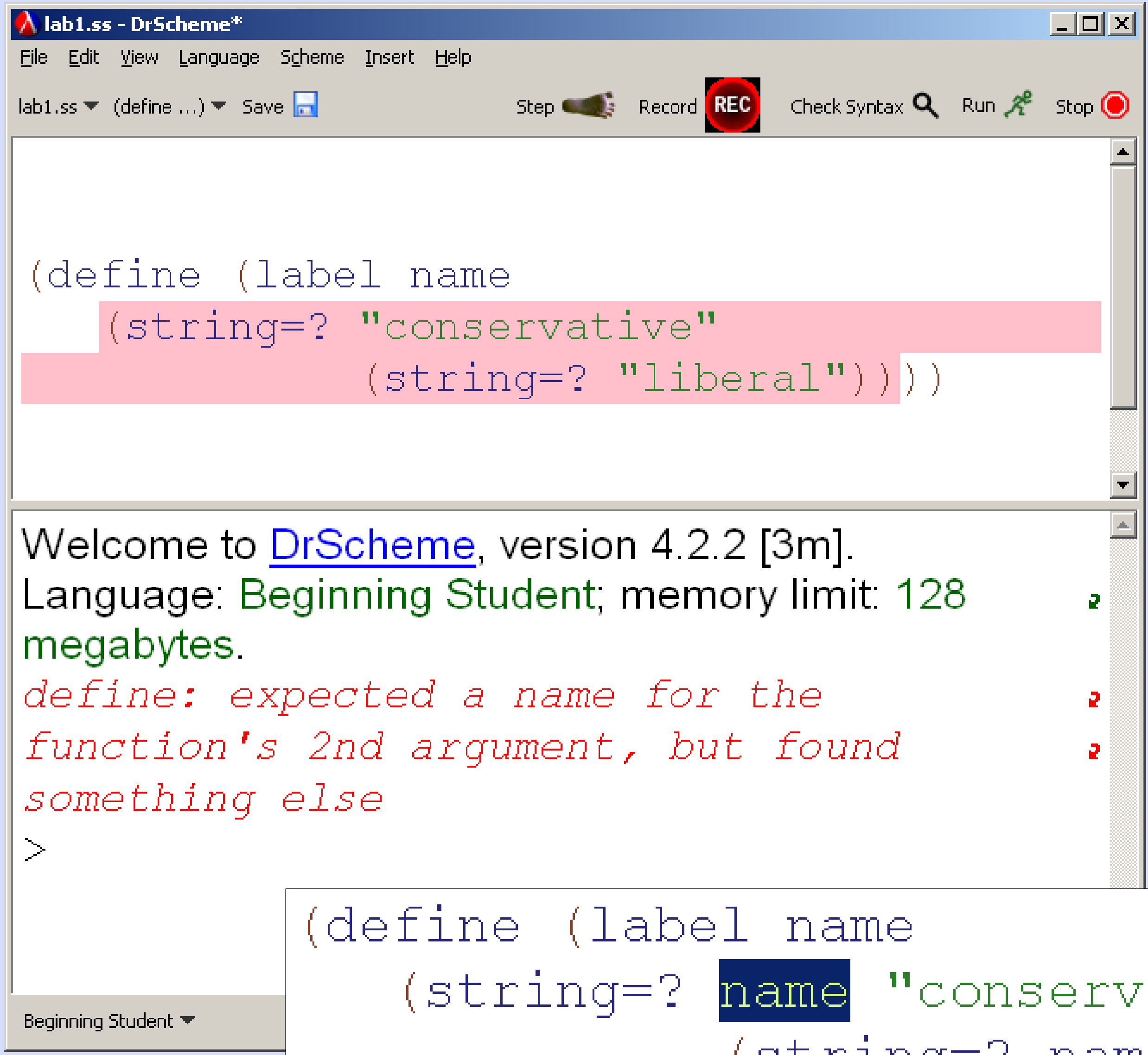
OK

How well do error messages support learning (or fail to?)

When errors fail to teach, in which ways do they fail?

What makes a good error message?
What is a valid metric of quality?

Can we make recommendations to the creators of pedagogical IDEs/compilers/languages?


lab1.ss - DrScheme*
File Edit View Language Scheme Insert Help
lab1.ss (define ...) Save
Step Record REC Check Syntax Run Stop
Beginning Student

```
(define (label name
    (string=? "conservative"
              (string=? "liberal"))))
```

```
Welcome to DrScheme, version 4.2.2 [3m].
Language: Beginning Student; memory limit: 128
megabytes.
define: expected a name for the
function's 2nd argument, but found
something else
>
```

```
(define (label name
    (string=? name "conservative"
              (string=? name "liberal"))))
```

```
;; string-one-of? string string string string -> boolean
(define (string-one-of? check-for-match stringOne stringTwo stringThree)
   cond [(and (string=? check-for-match stringOne))]
        [(and (string=? check-for-match stringTwo))]
```

define: expected only one expression for the function body, but found at least one extra part

```
(define (string-one-of? check-for-match stringOne stringTwo stringThree)
   cond [(string=? check-for-match stringOne)]
        [(and (string=? check-for-match stringTwo))]
        [(and (string=? check-for-match stringThree))])
```

```
(define (string-one-of? check-for-match stringOne stringTwo stringThree)
   cond [and ((string=? check-for-match stringOne))]
        [(and (string=? check-for-match stringTwo))]
        [(and (string=? check-for-match stringThree))])
```

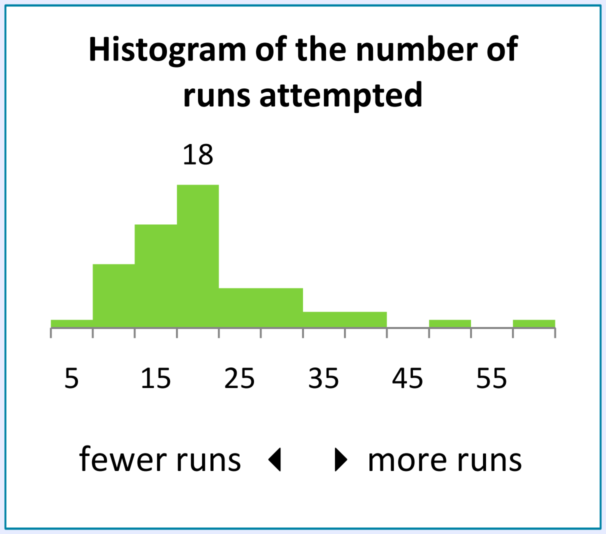
```
(define (string-one-of? check-for-match stringOne stringTwo stringThree)
   cond [(string=? check-for-match stringOne)]
        [(string=? check-for-match stringTwo)]
        [(string=? check-for-match stringThree)])
```

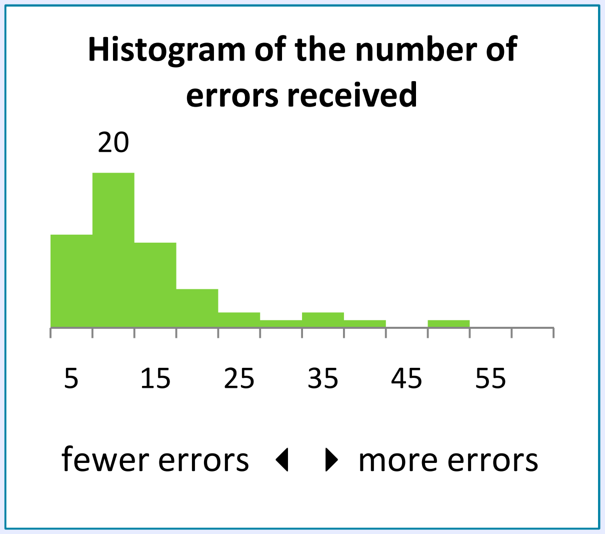
```
(define (string-one-of? check-for-match stringOne stringTwo stringThree)
   cond [(string=? check-for-match)]
        [(string=? check-for-match stringTwo)]
        [(string=? check-for-match stringThree)])
```

Read ▶ Understand ▶ Formulate

[DEL]	Deletes the problematic code wholesale.
[UNR]	Edit unrelated to the error message, and does not help.
[DIFF]	Edit unrelated to the error message, but it correctly addresses a different error or makes pro
[PART]	
[FIX]	

$$\kappa = \frac{\text{Agreement} - \text{Expected Agreement}}{1 - \text{Expected Agreement}}$$





For student s and category c, we compute:

$$p_{s,c} = \frac{[FIX]}{[UNR] + [PART] + [FIX]}$$

Then we take the unweighted average across the n students who are represented in the selected samples:

$$p_c = \left(\sum p_{s,c}\right)/\mathrm{n}$$

Coding Results for Lab #1

Category	Number presented	Number coded	DEL	UNR	DIFF	PART	FIX	p_c
paren. matching	129	26	0	3	1	3	19	76%
unbound id.	73	33	1	3	2	2	25	84%
syntax / define	73	32	2	11	4	4	11	50%
syntax / func. call	63	29	1	10	2	7	9	36%
syntax / cond	61	31	2	12	0	4	13	49%
arg. count	24	21	1	5	0	8	7	52%

Interviews

Four interviews

One hour long each

Done around the midterm

Average-to-good students

Observation From Interviews

1 of 2

Interviewer:	The error message says “the function body.” Do you know what “function body” means?
Student:	Nah, the input, everything that serves as a piece of input?
Interviewer:	Actually, it's this. When DrScheme says “function body” it means this part.
Student:	Oh man! I didn't…

[The student proceeds to fix the error successfully]

What DrScheme Says:

define: expected only one <u>expression</u> for the <u>function</u> <u>body</u>, but found at least one extra <u>part</u>.

What the Student Sees:

define: expected only one <u>rimagole</u> for the <u>blah's foo</u>, but found one extra <u>whatchamacallit</u>.

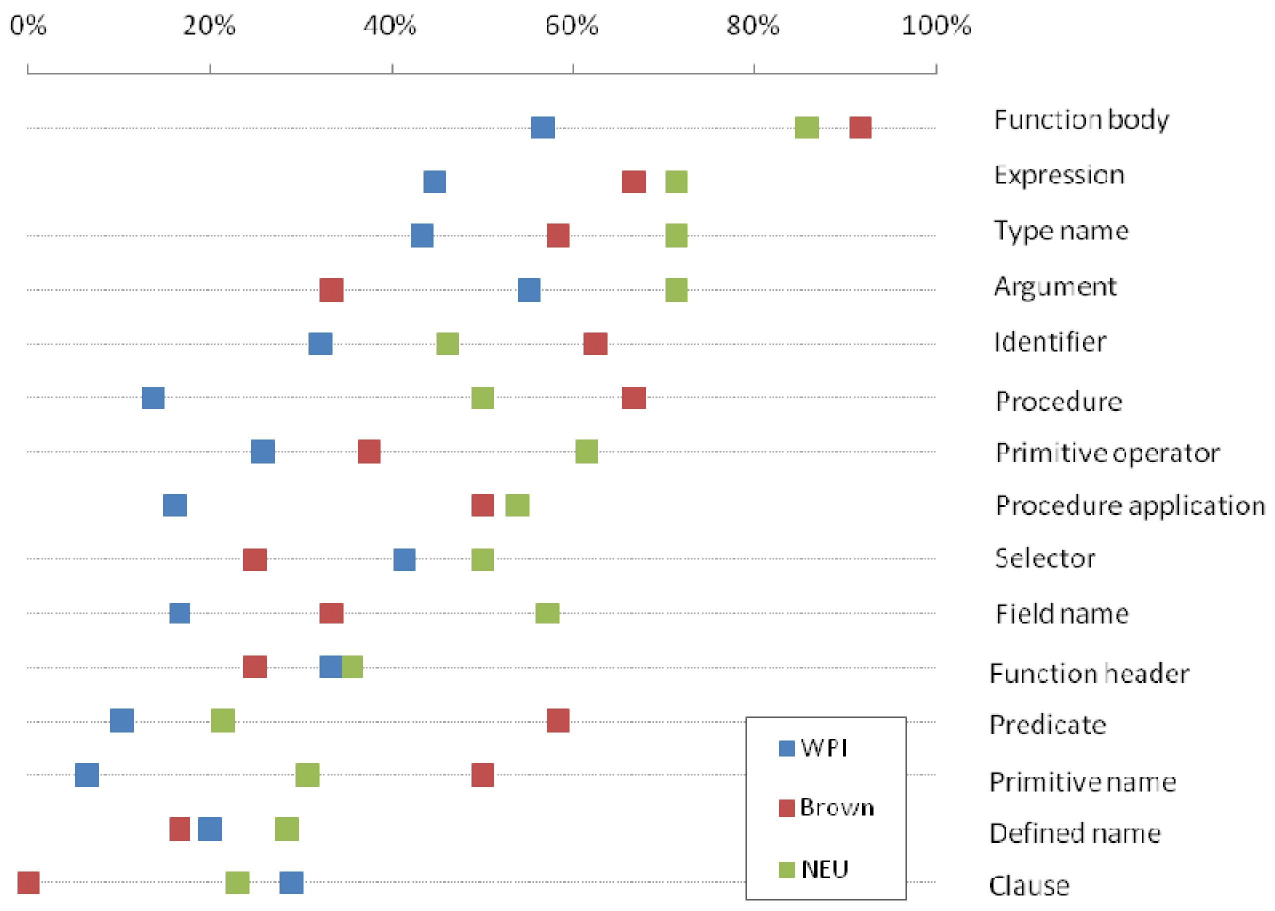
Circle <u>one</u> instance of each vocabulary term in the code below.

	Vocabulary term	**Sample usage**
Q1.	Argument	>: expects at least 2 <u>arguments, given 1</u>
Q2.	Selector	this selector expects 1 argument, here it is provided 0 arguments
Q3.	Procedure	this procedure expects 2 arguments, here it is provided 0 arguments
Q4.	Expression	expected at least two expressions after `and', but found only one expression
Q5.	Predicate	this predicate expects 1 argument, here it is provided 2 arguments

```
;; (make-book number string string number number bst bst)
(define-struct book (isbn title author year copies left right))

;; this-edition?:  bst number number -> boolean
;; Consumes a binary search tree, an ISBN number, and a year, and produces true
;; if the book with the given ISBN number was published in the given year
(define (this-edition? a-bst isbn-num year)
  (cond [(symbol? a-bst) false]
        [(book? a-bst)
         (cond [(= isbn-num (book-isbn a-bst))
                (= year (book-year a-bst))]
               [(< isbn-num (book-isbn a-bst))
                (this-edition? (book-left a-bst) isbn-num year)]
               [else (this-edition? (book-right a-bst) isbn-num year)])]))
```

Quiz Results



Average percent correct of that word on the quiz

Serendipitous Controlled Trials

✓ = Used in Class

	Brown	NEU	WPI
Primitive name	✓		
Procedure	✓		
Primitive operator	✓		
Field name		✓	
Procedure application	✓	✓	
Predicate	✓	✓	
Defined name			✓
Type name			✓
Identifier	✓		✓
Function body		✓	✓
Function header		✓	✓
Argument	✓	✓	✓
Clause	✓	✓	✓
Expression	✓	✓	✓
Selector	✓	✓	✓

	Estimate	P-Value
1	41.6%	0.000036
used	13.8%	0.014725
word[Argument]	-13.6%	0.250289
word[Clause]	-49.5%	0.000208
word[Defined name]	-36.0%	0.002968
word[Expression]	-5.9%	0.612732
word[Field name]	-22.0%	0.056020
word[Function body]	11.1%	0.344644
word[Function header]	-30.9%	0.009886
word[Identifier]	-15.3%	0.180584
word[Predicate]	-32.3%	0.007450
word[Primitive name]	-28.6%	0.014935
word[Primitive operator]	-16.1%	0.155599
word[Procedure]	-14.2%	0.207702
word[Procedure application]	-22.3%	0.055605
word[Selector]	-28.1%	0.021992
univ[brown]	13.5%	0.011538
univ[neu]	20.9%	0.000235

95% conf. interval:
[2.93%, 24.7%]

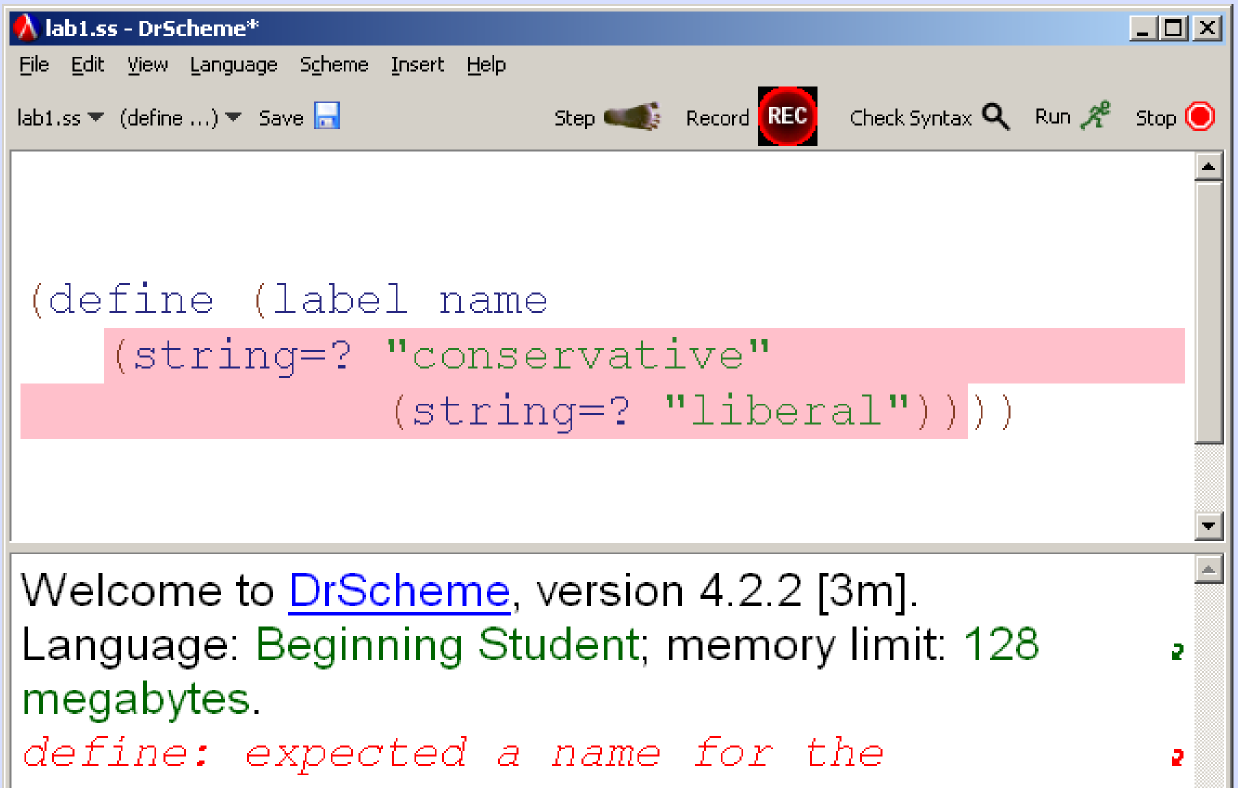
Old term	New term
Procedure Primitive name, Primitive operator Predicate Selector Constructor	Function
Name Identifier Argument Defined name	Variable, argument *(“argument” is reserved for actual arguments in function calls)*
Sequence	At least one
Structure type name	Structure name
Question—answer clause	A clause is expected to have a question and an answer
Function header Primitive name Keyword Type < >	*These words and notations are removed entirely and reworded in terms of other vocabulary words.*
Function body Expression Field name Type name Top level Binding Clause Part	*These words stay unchanged*

Observation From Interviews

2 of 2

Interviewer: When you get these highlights, what do they mean to you?

Student #1: The problem is between here and here, fix the problem between these two bars.


lab1.ss - DrScheme*
File Edit View Language Scheme Insert Help
lab1.ss (define ...) Save
Step Record REC Check Syntax Run Stop

```
(define (label name
    (string=? "conservative"
              (string=? "liberal"))))
```

Welcome to DrScheme, version 4.2.2 [3m].
Language: Beginning Student; memory limit: 128 megabytes.
define: expected a name for the

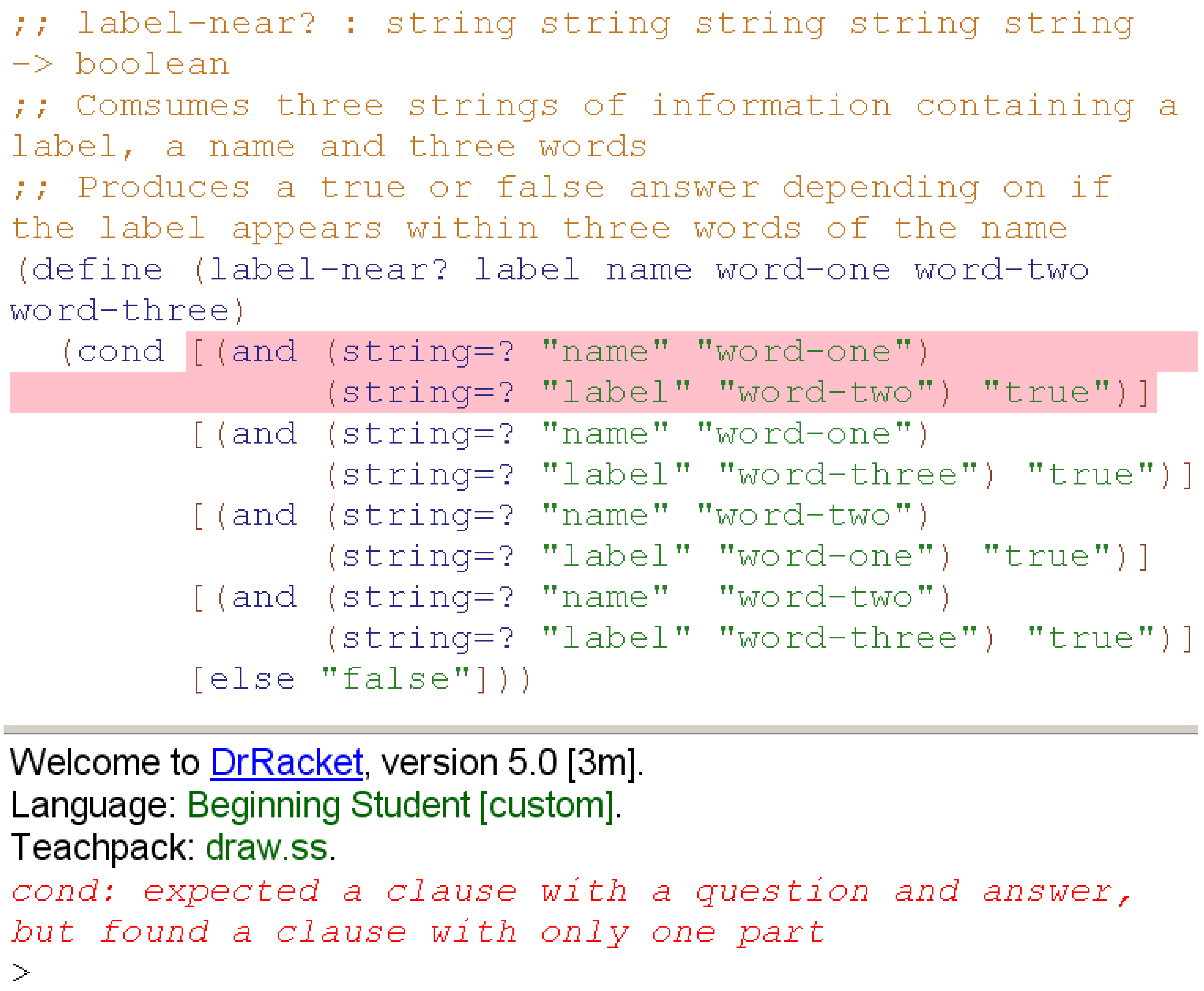
Interviewer:	You were saying that you pattern match on the highlight and don't read the messages at all.
Student #2:	I think that in the beginning it was more true, because the highlight were more or less “this is what's wrong,” so when I was a beginning programmer that's what I saw and that's what I would try to fix.

Interviewer:	When DrScheme highlights something, what does it highlight?
Student #3:	It highlights where the error occurred.
Interviewer:	Do you usually look for fixes inside the highlight?
Student #3:	mmm... I think I did at the beginning.

Interviewer:	Which one was more useful, the highlight or the message?
Student #2:	mmm... I would say the message. Because then highlight was redirecting me to here, but it didn't see anything blatantly wrong here. So I read the error message, which said that it expected five arguments instead of four, so then I looked over here.
Interviewer:	Would you say the highlight was misleading?
Student #2:	Yeah. Because it didn't bring me directly to the source.

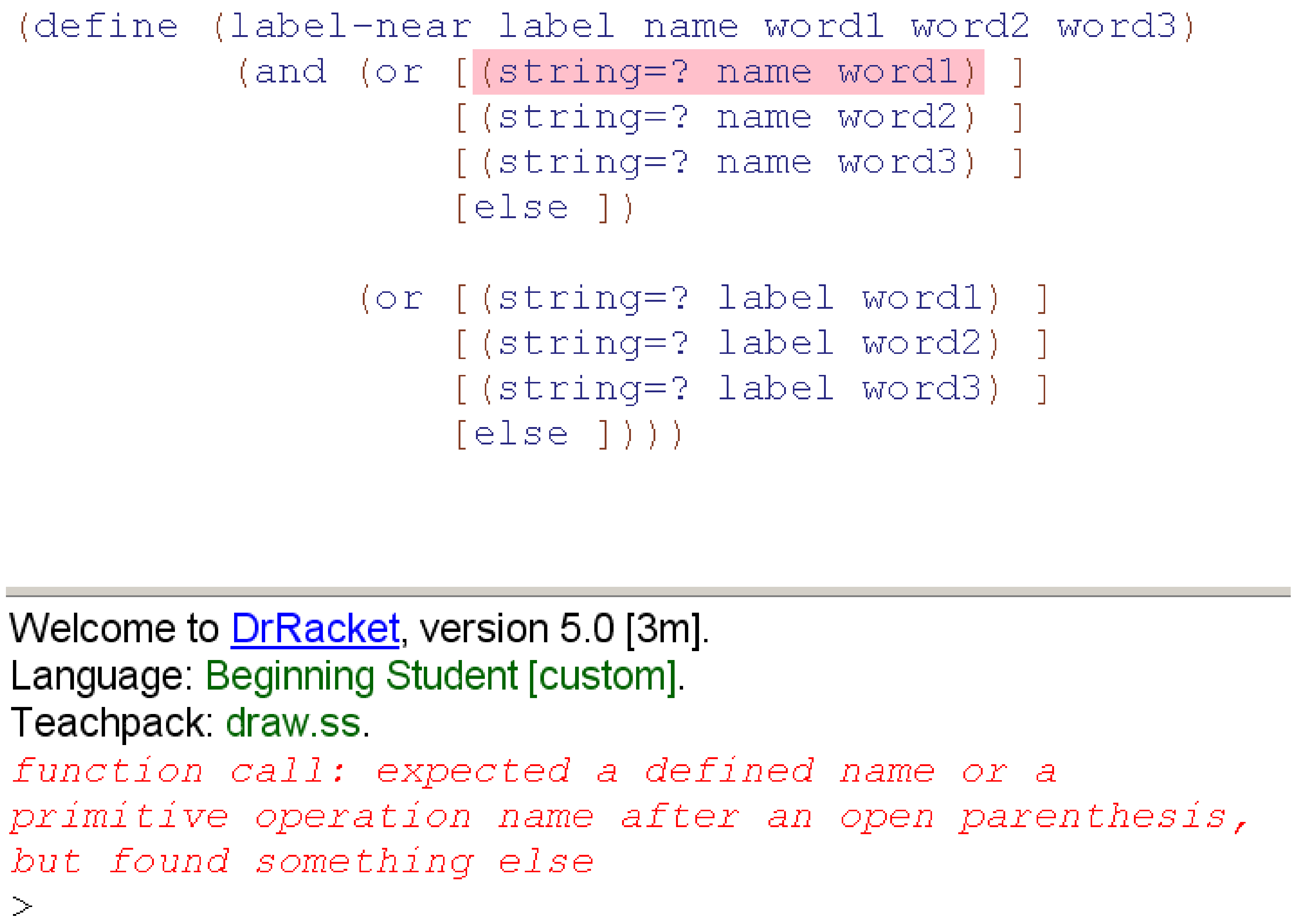
DrScheme’s Highlight Semantics

1. This expression contains the error
2. The parser didn’t expect to find this
3. The parser expected to see something after this, but nothing is there
4. This parenthesis is unmatched
5. This expression is inconsistent with another part of the code



```
;; label-near? : string string string string string
-> boolean
;; Comsumes three strings of information containing a
label, a name and three words
;; Produces a true or false answer depending on if
the label appears within three words of the name
(define (label-near? label name word-one word-two
word-three)
  (cond [(and (string=? "name" "word-one")
              (string=? "label" "word-two") "true")]
        [(and (string=? "name" "word-one")
              (string=? "label" "word-three") "true")]
        [(and (string=? "name" "word-two")
              (string=? "label" "word-one") "true")]
        [(and (string=? "name"  "word-two")
              (string=? "label" "word-three") "true")]
        [else "false"]))
```

```
Welcome to DrRacket, version 5.0 [3m].
Language: Beginning Student [custom].
Teachpack: draw.ss.
cond: expected a clause with a question and answer,
but found a clause with only one part
>
```



```
(define (label-near label name word1 word2 word3)
         (and (or [(string=? name word1) ]
                  [(string=? name word2) ]
                  [(string=? name word3) ]
                  [else ])

              (or [(string=? label word1) ]
                  [(string=? label word2) ]
                  [(string=? label word3) ]
                  [else ])))
```

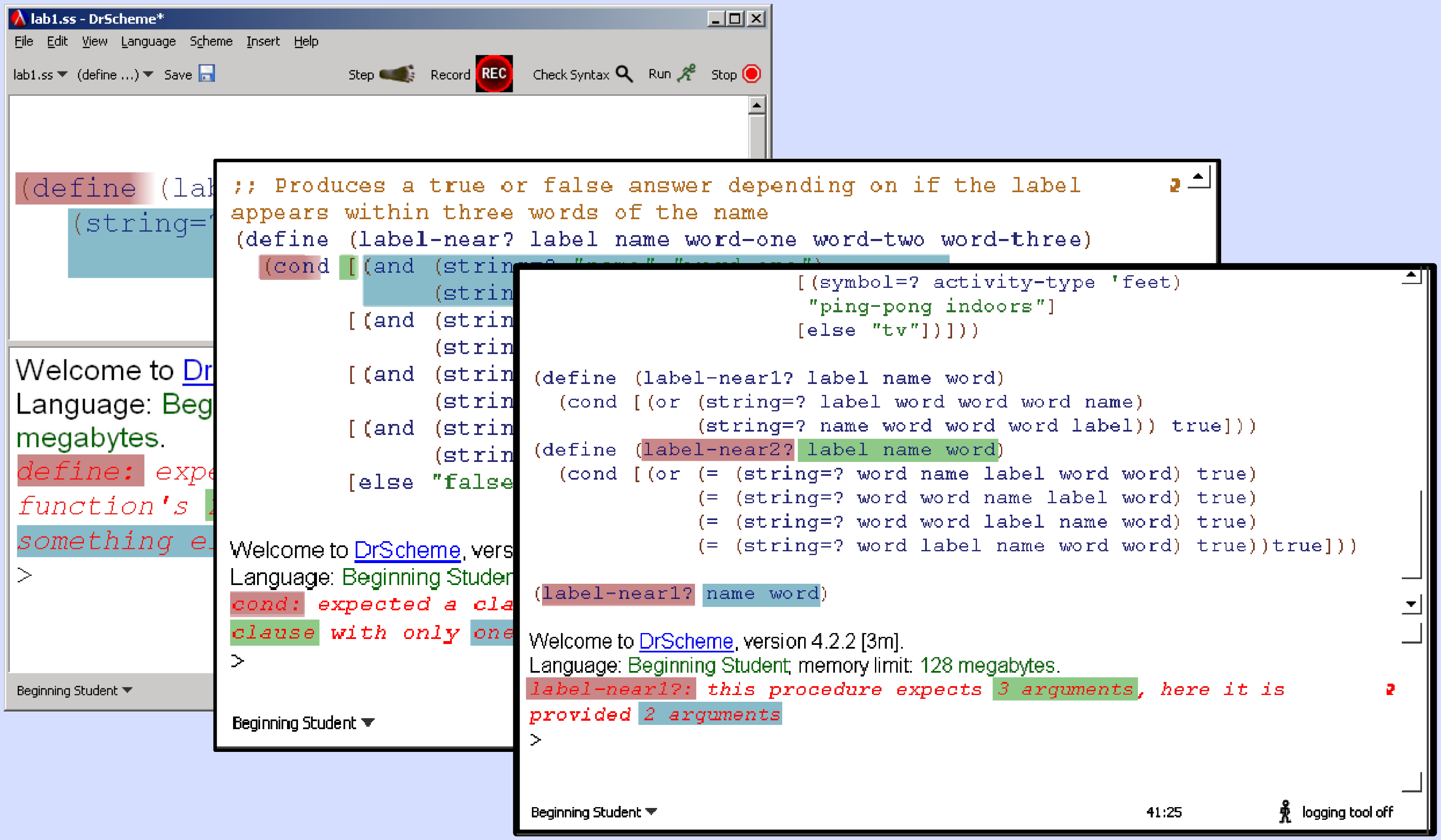
```
Welcome to DrRacket, version 5.0 [3m].
Language: Beginning Student [custom].
Teachpack: draw.ss.
function call: expected a defined name or a
primitive operation name after an open parenthesis,
but found something else
>
```

Summary of Findings

1. Error messages need to explicate the meaning of the highlight.

2. Students need an avenue through which they will learn the vocabulary.

3. Error messages are hard to get right;
user studies are important.


lab1.ss - DrScheme*
File Edit View Language Scheme Insert Help
lab1.ss (define ...) Save
Step Record REC Check Syntax Run Stop
Beginning Student

```
(define (lab
    (string=
```

```
Welcome to Dr
Language: Beg
megabytes.
define: expe
function's
something e
>
```

```
;; Produces a true or false answer depending on if the label
appears within three words of the name
(define (label-near? label name word-one word-two word-three)
  (cond [(and (strin
              (strin
        [(and (strin
              (strin
        [(and (strin
              (strin
        [(and (strin
              (strin
        [else "false
```

```
Welcome to DrScheme, vers
Language: Beginning Studen
cond: expected a cla
clause with only one
>
```

```
                          [(symbol=? activity-type 'feet)
                           "ping-pong indoors"]
                          [else "tv"])]))

(define (label-near1? label name word)
  (cond [(or (string=? label word word word name)
             (string=? name word word word label)) true]))
(define (label-near2? label name word)
  (cond [(or (= (string=? word name label word word) true)
             (= (string=? word word name label word) true)
             (= (string=? word word label name word) true)
             (= (string=? word label name word word) true))true]))

(label-near1? name word)
```

```
Welcome to DrScheme, version 4.2.2 [3m].
Language: Beginning Student; memory limit: 128 megabytes.
label-near1?: this procedure expects 3 arguments, here it is
provided 2 arguments
>
```

The End

Guillaume Marceau, Kathi Fisler (WPI)
Shriram Krishnamurthi (Brown)

gmarceau@wpi.edu