```
Josephs-MacBook-Air-2:scheme JosephHayes$ ls
__pycache__      ok          scheme_builtins.py   tests.scm
buffer.py        proj04.ok        scheme_reader.py ucb.py
images           questions.scm        scheme_tokens.py
mytests.rst      scheme.py        tests
Josephs-MacBook-Air-2:scheme JosephHayes$ atom scheme.py
Josephs-MacBook-Air-2:scheme JosephHayes$ python3 ok -q eval_apply -u
=====================================================================
Assignment: Project 4: Scheme Interpreter
OK, version v1.13.11
=====================================================================

Checking for software updates...
OK is up to date
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "ok/__main__.py", line 46, in <module>
  File "ok/client/cli/ok.py", line 188, in main
  File "ok/client/api/assignment.py", line 24, in load_assignment
  File "ok/client/sources/common/core.py", line 187, in __call__
  File "ok/client/api/assignment.py", line 153, in post_instantiation
  File "ok/client/api/assignment.py", line 202, in _load_tests
  File "ok/client/sources/ok_test/__init__.py", line 51, in load
  File "ok/client/sources/common/core.py", line 187, in __call__
  File "ok/client/sources/ok_test/models.py", line 45, in post_instantiation
  File "ok/client/sources/common/core.py", line 170, in __call__
  File "ok/client/sources/ok_test/doctest.py", line 20, in __init__
    Then running the module as a script will cause the examples in the
  File "ok/client/sources/common/interpreter.py", line 156, in __init__
  File "ok/client/sources/ok_test/scheme.py", line 41, in load
    def scheme_apply(procedure, args, env):
  File "/Users/JosephHayes/Desktop/61A/Projects/scheme/scheme.py", line 626, in create_global_frame
    BuiltinProcedure(scheme_eval, True, 'eval'))
  File "/Users/JosephHayes/Desktop/61A/Projects/scheme/scheme.py", line 80, in define
    self.value = self.symbol
AttributeError: 'Frame' object has no attribute 'symbol'
Josephs-MacBook-Air-2:scheme JosephHayes$ python3 ok -q eval_apply -u
=====================================================================
Assignment: Project 4: Scheme Interpreter
OK, version v1.13.11
=====================================================================

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Unlocking tests

At each "? ", type what you would expect the output to be.
Type exit() to quit

---------------------------------------------------------------------
Understanding scheme.py > Suite 1 > Case 1
(cases remaining: 6)

Q: A Scheme expression can be either...
Choose the number of the correct choice:
0) A special form or a call expression
1) A primitive expression or a combination
2) A primitive expression or a special form
3) A pair or a list
? 3
-- Not quite. Try again! --

Choose the number of the correct choice:
0) A special form or a call expression
1) A primitive expression or a combination
2) A primitive expression or a special form
3) A pair or a list
? 0
-- Not quite. Try again! --

Choose the number of the correct choice:
0) A special form or a call expression
1) A primitive expression or a combination
2) A primitive expression or a special form
3) A pair or a list
? 1
-- OK! --

---------------------------------------------------------------------
Understanding scheme.py > Suite 1 > Case 2
(cases remaining: 5)
```

```
Q: What expression in the body of scheme_eval finds the value of a name?
Choose the number of the correct choice:
0) SPECIAL_FORMS[first](rest, env)
1) scheme_symbolp(expr)
2) env.find(name)
3) env.lookup(expr)
? 1
-- Not quite. Try again! --

Choose the number of the correct choice:
0) SPECIAL_FORMS[first](rest, env)
1) scheme_symbolp(expr)
2) env.find(name)
3) env.lookup(expr)
? 2
-- Not quite. Try again! --

Choose the number of the correct choice:
0) SPECIAL_FORMS[first](rest, env)
1) scheme_symbolp(expr)
2) env.find(name)
3) env.lookup(expr)
? 3
-- OK! --

----------------------------------------------------------------------
Understanding scheme.py > Suite 1 > Case 3
(cases remaining: 4)

Q: How do we know if a given combination is a special form?
Choose the number of the correct choice:
0) Check if the first element in the list is a symbol
1) Check if the first element in the list is a symbol and that the
   symbol is in the dictionary SPECIAL_FORMS
2) Check if the expression is in the dictionary SPECIAL_FORMS
? 1
-- OK! --

----------------------------------------------------------------------
Understanding scheme.py > Suite 1 > Case 4
(cases remaining: 3)

Q: When and how do we create new Frames?
Choose the number of the correct choice:
0) Whenever a primitive or user-defined procedure is called; we use
   the apply method in subclasses of Procedure
1) Whenever a user-defined procedure is called; we use the
   make_call_frame method of LambdaProcedure
2) Whenever a new procedure is defined; we use the make_child_frame
   method in Frame
3) Whenever a primitive or user-defined procedure is called; we use
   the make_call_frame method of LambdaProcedure
? 1
-- OK! --

----------------------------------------------------------------------
Understanding scheme.py > Suite 1 > Case 5
(cases remaining: 2)

Q: What is the difference between applying builtins and applying user-defined procedures?
(Choose all that apply)

I.   User-defined procedures open a new frame; builtins do not
II.  Builtins simply execute a predefined function; user-defined
     procedures must evaluate additional expressions in the body
III. Builtins have a fixed number of arguments; user-defined procedures do not

---
Choose the number of the correct choice:
0) II only
1) I and III
2) II and III
3) III only
4) I and II
5) I, II and III
6) I only
? 4
-- OK! --

----------------------------------------------------------------------
Understanding scheme.py > Suite 1 > Case 6
(cases remaining: 1)

Q: What exception should be raised for the expression (1)?
```

```
Choose the number of the correct choice:
0) SchemeError("unknown identifier: 1")
1) SchemeError("malformed list: (1)")
2) AssertionError
3) SchemeError("1 is not callable")
? 3
-- OK! --

----------------------------------------------------------------------
OK! All cases for Understanding scheme.py unlocked.

Backup... 100% complete
Backup successful for user: joseph.hayes@berkeley.edu

Checking for software updates...
OK is up to date
Josephs-MacBook-Air-2:scheme JosephHayes$
```