Spring, 1992 Name: _____

CS 342 — Principles of Programming Languages

Smalltalk Test (Terms, Version 1)

The usual instructions apply.

Don't feel obligated to fill up all the space on the page. Conversely, don't feel limited by the space on the page, since you can use the back if you explicitly tell us on the front when you do so.

1 Terms

1. (4 points) What is "self" in a Smalltalk method? How is "self" used in Smalltalk?

2. (6 points) What is an object? Give an example, and state the importance of objects in Smalltalk.

3. (8 points) Describe and illustrate (i.e., with an example) what inheritance means for instance variables in Smalltalk.

4. (10 points) Consider the following class definitions.

```
(class Boolean Object
  ()
  (define ifTrue:ifFalse: (e1 e2) #subClassResponsibility)
  (define ifFalse:ifTrue: (e1 e2) (ifTrue:ifFalse: self e2 e1))
  (define and: (b) (ifTrue:ifFalse: self b (mkFalse)))
  (define or: (b) (ifFalse:ifTrue: self b (mkTrue)))
)
(class True Boolean
  ()
  (define initTrue () self)
  (define ifTrue:ifFalse: (e1 e2) e1)
  (define print () (begin (print #true) self))
)
(define mkTrue () (initTrue (new True)))
(class False Boolean
  ()
  (define initFalse () self)
  (define ifTrue:ifFalse: (e1 e2) e2)
  (define print () (begin (print #false) self))
)
(define mkFalse () (initFalse (new False)))
```

Describe the execution of the following expression, and give the output of that expression (both that explicitly called for and output from the read-eval-print loop).

```
(begin
  (set t (mkTrue))
  (print (or: (mkFalse) t)))
```

Your description of the execution of this expression must be a detailed trace of each function, value-op, and method called. You must note, for each method called, the classs of the method is executed. For example, "... then we call the `foo` method of class `Bar` ..." If the expression does not terminate, say that and explain why.