CSE 341 — Smalltalk Discussion Questions

1. The class Collection, which is at the root of the collection class hierarchy, includes a method `do:` for iterating through the collection. Using `do:`, define a `printOn:` method for Collection.

2. Define an `xor:` method for booleans. (Actually one is already built in, but ignore that fact . . .)

3. True or false? Smalltalk is type safe, but not statically type checked.

4. Suppose that we evaluate the following Smalltalk code.

```
| a b c |
a := Array new: 2.
b := Array new: 2.
c := a.

a at: 1 put: 10.
a at: 2 put: 20.

b at: 1 put: 30.
b at: 2 put: 20.

a=b ifTrue: [Transcript show: 'a and b are equal'.  Transcript cr].
a==b ifTrue: [Transcript show: 'a and b are identical'.  Transcript cr].
a=c ifTrue: [Transcript show: 'a and c are equal'.  Transcript cr].
a==c ifTrue: [Transcript show: 'a and c are identical'.  Transcript cr].

Transcript show: 'changing an element ...'.  Transcript cr.
c at: 1 put: 30.

a=b ifTrue: [Transcript show: 'a and b are equal'.  Transcript cr].
a==b ifTrue: [Transcript show: 'a and b are identical'.  Transcript cr].
a=c ifTrue: [Transcript show: 'a and c are equal'.  Transcript cr].
a==c ifTrue: [Transcript show: 'a and c are identical'.  Transcript cr].
```

What is printed to the transcript?

5. Suppose we are simulating trains in Smalltalk, and define a class Engine, as follows.

```
Object subclass: #Engine
  instanceVariableNames: 'idNumber'
  classVariableNames: ''
  poolDictionaries: ''


setIdNumber: n
  "set the engine's identification number (each engine should have one)"
  idNumber := n

kind
  "return a descriptive string saying what kind of engine this is"
  ^ 'engine'

printOn: aStream
  aStream nextPutAll: self kind.
  idNumber printOn: aStream.
```

One subclass of Engine is DieselEngine, defined as follows:

```
Engine subclass: #DieselEngine
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''

kind
  ^ 'diesel locomotive '
```

Another subclass of Engine is SteamEngine, defined as follows:

```
Engine subclass: #SteamEngine
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''

kind
  " , (comma) is the string concatenation message "
  ^ 'steam ' , super kind
```

Suppose we have made instances of the three classes as follows:

```
a := Engine new.        a setIdNumber: 8.
b := DieselEngine new. b setIdNumber: 17.
c := SteamEngine new.  c setIdNumber: 97.
```

What is printed to the transcript for each of the following messages?

```
a printOn: Transcript.
b printOn: Transcript.
c printOn: Transcript.
```