CSE 341 Midterm — May 5, 2000

Your name:

This exam is open book, open notes. You are to work independently. 80 points total.

1. (6 points) Write a Scheme expression to pick out the atom `squid` in the following lists (which will each be bound to `x`). You can use an expression you want, as long as there are no side effects or definitions of new global variables.

   (a) `(define x '(octopus shrimp squid clam))`

   (b) `(define x (cons shrimp (cons squid (cons smelt nil))))`

   (c) `(define x '( ((octopus)) ((shrimp)) ((squid)) ((clam)) ))`

   Hint: as an example, if the list is

   ```
   (define x '((squid clam) cow))
   ```

   then a correct answer is

   ```
   (caar x)
   ```

   Another correct (but more complicated) answer is

   ```
   (let ((seastuff (car x)))
      (car seastuff))
   ```

2. (6 points) Zero or more of the following Perl expressions are true. Circle all true expressions.

   (a) `"spring"`

   (b) `"spring" =~ /^[aeiou]*.*/`

   (c) `"spring" =~ /^[aeiou].*/`

3. (6 points) What does the following Perl code fragment print?

   ```
   $x = "the wonderful thing about Tiggers is Tiggers are wonderful things";
   $x =~ s/on(.*)ge//;
   print "$1\n$x";
   ```

4. (12 points) Consider the following Java class definitions. (These compile correctly.)

```
class Appliance {
  private static int applianceTag = 0;

  public int tag() {
    return applianceTag;
  }

  public void setTag(int t) {
    applianceTag = t;
  }

 public void printDescription() {
    System.out.println("a generic appliance");
  }
}


class Stove extends Appliance {

 public void printDescription() {
    System.out.println("a stove");
    super.printDescription();
  }
}


class SelfCleaningStove extends Stove {

  private int myTag = 0;

  public int tag() {
    return myTag;
  }

  public void setTag(int t) {
    myTag = t;
  }

 public void printDescription() {
    System.out.println("a self cleaning stove");
    super.printDescription();
  }
}
```

Now suppose we also have a class `Kitchen` with a `main` method. What is the result of compiling and executing the Java program for each of the following versions of `Kitchen` and `main`? The result might be that the program runs correctly, or it might have a compile time error, or a runtime error. If the program compiles correctly and can be run, give the output. Otherwise explain what the error is.

(a)

```
public class Kitchen {
  public static void main (String [ ] args) {
    Appliance a = new Appliance();
    Appliance b = new Stove();
    Appliance c = new SelfCleaningStove();
    a.printDescription();
    b.printDescription();
    c.printDescription();
  }
}
```

(b)

```
public class Kitchen {
  public static void main (String [ ] args) {
    SelfCleaningStove c = new SelfCleaningStove();
    Stove b = c;
    b.printDescription();
    c.printDescription();
  }
}
```

(c)

```
public class Kitchen {
  public static void main (String [ ] args) {
    Appliance a = new Appliance();
    Appliance b = new Stove();
    Appliance c = new SelfCleaningStove();
    a.setTag(100);
    b.setTag(200);
    c.setTag(300);
    System.out.println(a.tag());
    System.out.println(b.tag());
    System.out.println(c.tag());
  }
}
```

5. (10 points) Tacky but easy-to-grade true/false questions!

   (a) Java bytecodes are machine instructions for a virtual register-based machine that includes 32 general-purpose and 16 floating point registers.

   (b) As soon as an object in Java is no longer accessible from any live variable, its "finalize" method will be invoked and it will be garbage collected.

   (c) Scheme is type safe, but not statically type checked.

   (d) Java is type safe, but not statically type checked.

   (e) Perl's type declarations, along with its static type checking, mean that type errors in a Perl program are discovered at compile time.

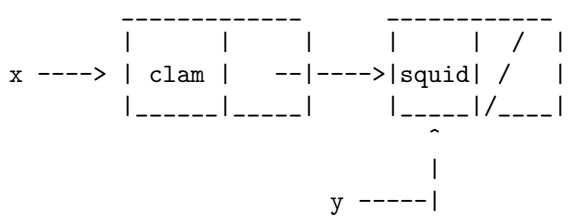
6. (8 points) Draw a box-and-arrow diagram that shows the list structure and bindings of variables after the following Scheme expressions have been evaluated:

```
(define x '(a b c))
(define y (append (cons 1 nil) x))
(define z (cdr y))
```

   Hint: for the expressions

```
(define x '(clam squid))
(define y (cdr x))
```

   you would draw the following:



7. (12 points) What is the result of evaluting the following Scheme expressions?

   (a)
```
(* (+ 3 4) (- 6 2))  ; an easy one to start off ...
```

   (b)
```
(cddr '(a (b)))
```

   (c)
```
(let ((x '(a b))
      (y '(1 2))
  (let ((x '(c d))
        (y '(3 4 5))
        (f (lambda (z) (append x z))))
    (f x))))
```

   (d)
```
(let ((a cons)
      (b 3)
      (c '(+ 1 2)))
  (eval (list a b c) user-initial-environment))
```

8. (20 points) Java includes 8 primitive types (which aren't objects). What are the advantages and disadvantages of having these 8 primitive types separate from ordinary objects, as opposed to making everything be an object? Consider the regularity and usability of the language, efficiency of runtime storage management, potential for efficiency of code generated by a JIT, and any other issues you think are relevant.