Programming Paradigms (COMP 3007)
Midterm Exam II

Carleton University
School of Computer Science
Fall 2012

Name: _____ **Student #:** _____

1. [20%] (a) Suppose we say

```
(define baz 10)
(define s (make-serializer))

(parallel-execute
         (s (lambda () (set! baz (/ baz 2))))
         (s (lambda () (set! baz (+ baz baz)))))
```

What are the possible values of `baz` **after this finishes?**

Answer → 10

(b) Now suppose that we change the example to leave out the serializer, as follows:

```
(define baz 10)

(parallel-execute
         (lambda () (set! baz (/ baz 2)))
         (lambda () (set! baz (+ baz baz))))
```

What are all of the possible values of `baz` **this time?**

Answers → 10, 5, 15, 20

2. **[25%] Suppose we want to write a procedure** `prev` **that takes as its argument a procedure** `proc` **of one argument.** `prev` **returns a new procedure that returns the value returned by the previous call to** `proc`**. The new procedure should return** `#f` **the first time it is called.**
   For example:

```
> (define slow-square (prev square))
> (slow-square 3)
#f
> (slow-square 4)
9
> (slow-square 5)
16
```

   Define procedure `prev` **in Scheme.**

```
(define (prev proc)
     (let ((old-result #f))
     (lambda (x)
          (let ((return-value old-result))
               (set! old-result (proc x))
               return-value))))
```

3. **[30%] Create a Prolog database, describing following relations:**

- **Coppola, Tarantino and Spielberg are directors**
- **Spielberg directed the following movies:**
  - **Name: Schindlers list, year: 1993, genre: drama**
  - **Name: AI, year: 2001, genre: sci-fi**
  - **Name: ET, year: 1982, genre: drama**
- **Spielberg and Tarantino have won the Oscars**
- **Any movie made after 1990 by an Oscar-winning director is worth watching**

And ask the following question in Prolog:

- **What the name of a movie worth watching?**

```
director('Coppola').

director('Spielberg').

director('Tarantino').

directed('Spielberg', movie('Schindlers list' , 1993, drama)).

directed('Spielberg', movie('AI', 2001, 'sci-fi')).

directed('Spielberg', movie('ET', 1982, drama)).

won_oscars('Spielberg').

won_oscars('Tarantino').

worth_watching(X) :-

          directed(D, movie(X,Y,_)), Y > 1990, won_oscars(D).


?- worth_watching(X).
```

4. **[25%] The Fibonacci numbers are the integer sequence**
   0, 1, 1, 2, 3, 5, 8, 13, 21, …
   in which each item is formed by adding the previous two. Define a predicate in Prolog called fib(N,F) where F is the Nth Fibonacci number.

```
fib(1, 0).
fib(2, 1).
fib(N, F) :-
     N1 is N – 1, fib(N1, F1),
     N2 is N - 2, fib(N2, F2),
     F is F1 + F2.
```