HW 1 – Evaluation and Environments
CS 421 – Fall 2012
Revision 1.1

Assigned September 1, 2012
Due September 11, 2012, 11:59 pm
Extension 48 hours (20% penalty)

1 Change Log

1.1 Program point `(* 6 *)` was misplaced. It has been moved back to be just after the declaration `let x = 1;;`.

1.0 Initial Release.

2 Objectives and Background

The purpose of this HW is to test your understanding of

- the order of evaluation of expressions in OCaml;

- the scope of variables, and the state of environments used during evaluation;

Another purpose of HWs is to provide you with experience answering non-programming written questions of the kind you may experience on the midterms and final.

3 Turn-In Procedure

Using your favorite tool(s), you should put your solution in a file named `hw1.pdf` (the same name as this file has on the website). If you have problems generating a pdf, please seek help from the course staff. Your answers to the following questions are to be submitted electronically via the handin script as though an MP. This assignment is named `hw1`.

4 Problem

1. (15 pts + 3 pts Extra Credit) Below is a fragment of OCaml code, with various program points indicated by numbers with comments.

```
1;;
1 = 1 + 1;;
2 = 1 + 1;;
(* 1 *)
let a = 10;;
let x = 11 + a;;
(* 2 *)
let f x y = x - y + a;;
(* 3 *)
let b = (f x a) in
          (let a = 5 in
(* 4 *)
              a + b + x );;
(* 5 *)
let x = 1;;
(* 6 *)
let h y = (let x = 2 in
            (let y = 20 in
              f x y) );;
(* 7 *)


(* extra credit *)
let h y = f (y+4) in
    (let f x y = x * y in
(* 8 *)
        (let a = 1 in
            (h 10) (f a a) ) );;
(* 9 *)
```

For each of program points 1, 2, 3, 4, 5, 6, and 7 please describe the environment in effect after evaluation has reached that point. Program points 8 and 9 are extra credits. You may assume that the evaluation begins in an empty environment, and that the environment is cumulative thereafter. The program points are supposed to indicate points at which all complete preceding declarations (including local ones) have been fully evaluated. You may name your environments and closures, and use their name in describing other environments, but all applications of the update operator $(+)$ should be expanded out.