**CSE340 FALL 2021 HOMEWORK 5**

**Due Monday Thursday December 2nd by 11:59 PM**

1. The homework has 2 problems
2. Your answers must be typed.
3. On Gradescope, you should submit the answers to separate question separately.

Unless otherwise notes, you do not need to show your work and the questions will be graded based on the answers, but remember that this homework is part of your preparation for the final exam

**Problem 1 (Static and Dynamic Scoping)**

Consider the following program written in a C-like syntax. Assume parameters are passed by value.

int a, b, c, d; // global variables **initially 0**

int g(int a; int b)

{

print(“%d %d %d %d”, a, b, c, d);

return a + b + c + d;

}

int f(int c)

{

int b;

b = 3;

b = g(a,b);

{ int c;

int d;

d = 4;

c = 8;

g(a,b);

}

g(a,b);

return b;

}

void main()

{

int a;

int b;

a = 4;

b = 5;

a = f(b);

g(b,a);

}

1. What is the output of this program if static scoping is used

2. What is the output of this program if dynamic scoping is used.

**Problem 2 (Hindley Milner Type checking)**

For this problem, you should give the answers and you do not need to show your work if there is no type checking error. **If there is a type checking error, you should show your work**. You can use an online OCaml editor to check your answers, but you should not solely rely on that. If you do, you will not do well on the final.

For each question, just give the type of x or explain why there is a type mismatch.

1. What is the type of x in:

let f1 x = 1 + x ;;

1. What is the type of x in:

let f2 a x = x 1 +. a ;;

1. What is the type of x in:

let f3 x y f = f 1 + f x.(y) ;;

1. What is the type of x in:

let f4 f x i = f 1.0 + x.(i) + f i ;;

1. What is the type of x in:

let f5 x y i = x (if y.(i) then i else i+1) = y.(i);;

1. What is the type of x in:

let f6 x y = y x + x (y x) ;;

1. What is the type of x in:

let f7 a b c = if b c then a c else b c ;;

1. What is the type of x in:

let f8 x b c = if b c then x c else b (x c) ;;

1. What is the type of x in:

let rec f9 x m = match x with

[] -> x

| h::t -> h\*m:: f8 m t;;

1. what does f9 calculate? You can try different examples using the OCaml command line
2. What is the type of x in:

let rec f10 x l = match l with

[] -> []

| h::t -> x h :: f10 x t;;

1. What does f10 f1 [1;2.3.4.5] do? f1 is the function f1 defined above
2. What is the type of x in:

let rec f11 x l = match (x,l) with

([],\_) -> []

| (\_,[]) -> []

| (h::t,\_) -> List.append (f10 h l) (f11 t l);;

1. What does f11 [ (fun n->n+1) ; (fun n -> n+2) ] [1;2;3] do?