**CSE340 FALL 2020 HOMEWORK 5**
**Due Monday  30 November 2020 by 11:59 PM**

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

## Problem 1 (Static and Dynamic Scoping)

Consider the following program written in a C-like syntax (well not quite C-like but should not be hard to understand). Assume parameters are passed by value.

```
a, b, x : int;       // global variables

int g(a : int; d: int)
{
        print a;
        print b;
        print x;
        print d;
        return a + b+ x + d;
}

int f(x : int)
{
        b : int;
        b = 3;
        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 (Structural, Name and Internal Name Equivalence)

Consider the following type declarations

```
TYPE
   T0  =    int;
   T1  =    real;
   T2  =    pointer to int;
   T3  =    pointer to real;
   T4  =    pointer to T0;
   T5  =    pointer to T1;
   T6  =    struct {
                  a: int;
            };
   T7  =    struct {
                  b: int;
                  a: pointer to T8;
            };
   T8  =    struct {
                  a: T0;
                  b: pointer to T7;
            };
   T9  =    struct {
                  x: T7 * T8 -> T7; // function of T7 and T8 that returns T7
                  y: T11 * T12 -> T8;
            };
   T10 =    struct {
                  x: T8 * T7 -> T8; // function of T7 and T8 that returns T8
                  y: T12 * T11 -> T7;
            };
   T11 =    array [4][5] of T9;
   T12 =    array [4][5] of T10
```

For each of the following types, list the types that are equivalent, assuming structural equivalence:

1. T0
2. T2
3. T4
4. T6
5. T8
6. T10
7. T12

```
Consider the following variable declarations (this is a continuation of the
previous declarations
```

**VARS**

```
   x    : pointer to int;
   y    : pointer int;
   z    : T3;
   p, q : T9 -> pointer to int;
   r    : T9 -> T3;
   s    : T9;
   t    : T10;
```

For each of the following assignment statements write if the statement is valid under structural equivalence, name equivalence, or internal name equivalence. Write all that apply.

1. x = y
2. y = z
3. p = q
4. q = r
5. z = p(s)
6. x = p(t)
7. z = r(t)
8. z = r(s)

## Problem 3 (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. 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 of the following determine the type of the function.

1. let f1 x = 1 ;;

2. let f2 x = x +. 1.0 ;;

3. let f3 x i = x.(x.(i)) ;;

4. let f4 x y i = y.(x.(i)) ;;

5. let f5 x y = x y ;;

6. let f6 x y = x (y+1) ;;

7. The following declaration results in a type checking error. Explain why

```
let rec f7 l1 l2 = match (l1,l2) with
            ([],_) -> l2
          | (_,[]) -> l1
          | (h1::t1,h2::t2) -> f6 h1 h2 :: f7 t1 t2 ;;
```

8. let rec f8 l1 l2 = match (l1,l2) with

```
            ([],_) -> l2
          | (_,[]) -> l1
          | (h1::t1,h2::t2) -> h1 + h2 :: f8 t1 t2 ;;
```

9. what does f8 calculate? You can try different examples using the OCaml command line