Todd Wenker
CSE 340
Professor Doupe
11/16/15

CSE 340 Fall 2015
HOMEWORK 4
Assigned 11/4/2015
Due 11/16/2015 by 11:59:59 pm on Blackboard
Remember that late submissions are not accepted for homework.
For all answers, show your work for partial credit.

All submissions should be typed. Exception can only be made for drawing parse trees, which can be handwritten and scanned in the submitted document.

Problem 1. Consider the following declarations

TYPE DECLARATIONS

```
T1 =    int;
T2 =    pointer to T1;
T3 =    pointer to int;
T4 =    pointer to string;
T5 =    struct {
                a : int;
                b : array of T6;
        }
T6 =    struct {
                a : T1;
                b :  array of T5;
        }
T7 =    struct {
                a: T5;
                b: array of T6;
        }
```

Determine which types are structurally equivalent.

T2, T3. Both pointers to structurally equivalent types (T1 which is of type int and int).
T5,T6. The first type of each struct are structurally equivalent and because we assume structurally equivalence of all types until proved otherwise, the array of T6 is structurally equivalent to the array of T5.

Problem 2.

Consider the following piece of C code

```
#include <stdio.h>
int a;

void f(int x, int y, int z)
{
      int b;

      x = x+1;
      y = y+1;
      b = x+y;
      z = z+1;
      a = b;
}

int main()
{
   int b = 1;
   int c[3] = {0,0,0};

   {
        int a = 0;
        f(a,b,c[b]);
   }

   printf("%d\n", a);
   printf("%d\n", b);
   printf("%d\n", c[0]);
   printf("%d\n", c[1]);
   printf("%d\n", c[2]);
   return 0;
}
```

1. What is the output of this program if parameters are passed by value?
   3
   1
   0
   0
   0
2. What is the output of this program if parameters are passed by reference?
   3
   2
   0
   1
   0
3. What is the output of this program if parameters are passed by name?
   3

2
0
0
1

Show your work for partial credit!

Problem 3.

Consider the following piece of C code

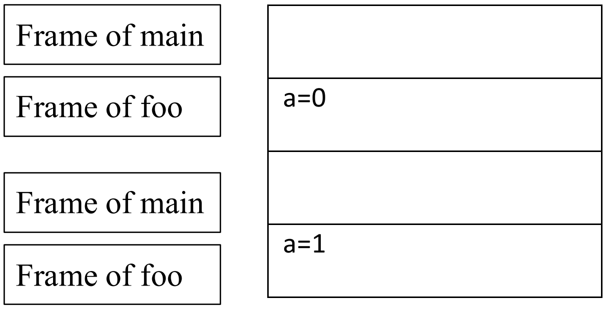
```
#include <stdio.h>

int main();
int x = 0;

int foo(int a)
{
   printf("In foo\n");
   x = x + 1;
   if (a < 1)
   {
        main();
   }
   // Location 1
   printf("Leaving foo\n");
   return a + 4;
}

int main()
{
   printf("In main\n");
   x = foo(x);
   printf("Leaving main\n");
   return 0;
}
```

1. Draw the program stack at the first execution of Location 1 (specified in the comment). Label on the stack each function frame, and inside each function frame label the parameters to the function, the values of those parameters, the function's local parameters, and the values of those local parameters. You do not need to follow precise cdecl calling convention. Assume static scoping and pass-by-value semantics.


Frame of main
Frame of foo
a=0
Frame of main
Frame of foo
a=1