ICS-365 HW5

\* Due Date – See Class Schedule \*

Note – there are two questions, do not skip question 2 on the next page

1. Consider the following C program:

**void** fun(**void**) {

**int** a, b, c; /\* definition 1A \*/

. . .

**while** (. . .) {

**int** b, c, d; /\*definition 2B \*/

. . . <------------- 1

**while** (. . .) {

**int** c, d, e; /\* definition 3C \*/

. . . <------------- 2

}

. . . <-------------- 3

}

. . . <---------------- 4

}

For each of the four marked points in this function, list each visible

variable, along with the number of the definition statement that defines it.

1. Definition 1: a

Definition 2: b,c,d

1. Definition 1: a

Definition 2: b

Definition 3: c,d,e

1. Definition 1: a

Definition 2: b,c,d

1. Definition 1: a,b,c

2. Consider the following skeletal C program:

**void** fun1(**void**); /\* prototype \*/

**void** fun2(**void**); /\* prototype \*/

**void** fun3(**void**); /\* prototype \*/

**void** main() {

**int** a, b, c;

. . .

}

**void** fun1(**void**) {

**int** b, c, d;

. . .

}

**void** fun2(**void**) {

**int** c, d, e;

. . .

}

**void** fun3(**void**) {

**int** d, e, f;

. . .

}

Given the following calling sequences and assuming that dynamic scoping

is used, what variables are visible during execution of the last function

called? Include with each visible variable the name of the function in

which it was defined.

1. main calls fun1; fun1 calls fun2; fun2 calls fun3.

**a-main, b-fun1, c-fun2, d e f-fun3**

1. main calls fun1; fun1 calls fun3.

**a-main, b c- fun1, d e f-fun3**

1. main calls fun2; fun2 calls fun3; fun3 calls fun1.

**a-main, b c d-fun1, e f-fun3**

1. main calls fun3; fun3 calls fun1.

**a-main, b c d-fun1, e f-fun3**

1. main calls fun1; fun1 calls fun3; fun3 calls fun2.

**a-main, c d e-fun2, b-fun1, f-fun3**

1. main calls fun3; fun3 calls fun2; fun2 calls fun1.

**a-main, b c d-fun1, f-fun3, e-fun2**