Tony Maldonado
CS 471
September 30, 2020

Chapter 5 Problem Set

Question 4

1. Dynamic type binding is closely related to implicit heap-dynamic variables. Explain this relationship.

   Answer:

   - Implicit heap-dynamic variables are closely related to dynamic type binding because implicit heap-dynamic variables use dynamic type binding. Dynamic type binding is the binding of a type to a variable at runtime, and implicit heap dynamic variables can only bind to a type at runtime when the value of the variable is assigned, so they're very closely related.

Question 9

2. Consider the following Python program:

```
x = 1;
y = 3;
z = 5;

def sub1():
        a = 7;
        y = 9;
        z = 11;
        …

def sub2():
        global x;
        a = 13;
        x = 15;
        w = 17;
        …
        def sub3():
                nonlocal a;
                a = 19;
                b = 21;
                z = 23;
                …
        …
```

   List all the variables, along with the program units where they are declared, that are visible in the bodies of sub1, sub2, and sub3, assuming static scoping is used.

Tony Maldonado
CS 471
September 30, 2020

Answer:

- In sub1():
  - a = 7 from sub1
  - y = 9 from sub1
  - z = 11 from sub1
  - x = 1 from main
- in sub2():
  - a = 13 from sub2
  - x = 15 from sub2
  - w = 17 from sub2
  - y = 3 from main
  - z = 5 from main
- in sub3():
  - a = 19 from sub3
  - b = 21 from sub3
  - z = 23 from sub3
  - x = 15 from sub2
  - w = 17 from sub2
  - y = 3 from main

Tony Maldonado
CS 471
September 30, 2020

Question 11

3. 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.

a. main calls fun1; fun1 calls fun2; fun2 calls fun3.
b. main calls fun1; fun1 calls fun3.
c. main calls fun2; fun2 calls fun3; fun3 calls fun1.
d. main calls fun3; fun3 calls fun1.
e. main calls fun1; fun1 calls fun3; fun3 calls fun2.
f. main calls fun3; fun3 calls fun2; fun2 calls fun1.

Answers:

a.) a: main
    b: fun1
    c: fun2
    d, e, f: fun3

Tony Maldonado
CS 471
September 30, 2020

b.) a: main
b, c: fun1
d, e, f: fun3

c.) a: main
b, c, d: fun1
e, f: fun3

d.) a: main
b, c, d: fun1
e, f: fun3

e.) a: main
b: fun1
c, d, e: fun2
f: fun3

f.) a: main
b, c, d: fun1
e: fun2
f: fun3