Nathaniel Daniel

Nicolescu

CS 326

28 September 2020

HW3

1.

   a. The variable's type and name are both assigned at compile time.
   b. The range of an integer is contained at implementation time. The range of an int is fixed by the C++ compiler on each platform.
   c. The definition of a `char` is fixed at design time as language designers must choose the names and semantics of their basic types of their language while they conceptualize it.
   d. The address of a local variable is a runtime binding as it is generally impossible to know its address until the program is executed.
   e. The address of a library function is defined at runtime as the memory locations of functions may change between different computers or program executions.
   f. The referencing environment is a runtime binding as the function cannot know its environment until it is called with arguments.
   g. The total memory needed is a runtime binding as memory can be allocated and deallocated continuously and unpredictably in a running program.

2. A language that does dynamic scoping cannot do type checking at compile time. The compiler can't know the type of the variables accessed until an access occurs at runtime

due to the nature of dynamic scoping. Yes. The compiler knows the type of every variable that is accessed at compile time.

3. Scheme uses Static Scoping. If Scheme was dynamically scoped, the following program's output would be 0 1, not 0 0.

   Test Program:

```
; HW3
; Nathaniel Daniel
; CS 326
; Load: (load "hw3.scm")
; Test: (test)
(define x 0)
(define (print-x)
        (display x)(newline)
)
(define (print-x-shadowed f)
        (define x 1)
         (print-x)
)
(define (test)
        (display "X: ")
        (print-x)
        (display "Shadowed X: ")
        (print-x-shadowed print-x)
```

'()

)

(test)

4.

Shallow Binding:

1 0 2 0 3 0 4 0

With shallow binding the local x is always accessed.

Deep Binding:

1 0 ? 2 0 0 4 4

set_x accesses the local variable when n is 1 or 3 and the global one when n is not 1 or 3. Similarly, print_x accesses the local variable when n is 1 or 2. Otherwise, it accesses the local one.