Question 1. //Chi-Hao Tu & Yaoyao Liu

(a) Symbol Tables with Static Scoping

P4:
a : int global
b : int parameter to f::int global
f : int -> int function

P5:
a : int global
b : int global
f : int -> int function
main : int -> int function

P6:
a : int global
b : int local to main::int global
f : int -> int function
main : int -> function

P7:
a : int local to main::int global
b : int local to main::int global
f : int -> int function
main : int -> function

P8:
a : int local to main::int global
b : int local to block in main::int global
f : int -> int function
main : int -> int function

P9:
a : int local to main
b : int local to block in main
f : int -> int function
main : int -> int function

(b) Output with Static Scoping

- 1.5
- 2.3
- 3.1
- 4.3
- 5.6

Question 2

(a) Symbol Tables with Dynamic Scoping

P3:
a : int = 6 local to main
b : int = 3 local to main
f : int -> int function
main : int -> int function

P4:
a : int = 6 local to main (from the previous declaration)
b : int = 1 local to the block
f : int -> int function
main : int -> int function

P5:
a : int = 6 local to main
b : int = 3 local to main
f : int -> int function
main : int -> int function

P6:
a : int = 3 global
b : int = 5 global
f : int -> int function

P7:
a : int = 3 global
b : int = 5 global
f : int -> int function

P8:
a : int parameter to f
b : int local to f (assigned within `f`, so it doesn't exist until `f` is called)
c : int local to f (assigned within `f`, so it doesn't exist until `f` is called)
f : int -> int function

P9:
a : int = 6 local to main
b : int = 6 local to main (after being assigned the result of `f(b)`)
f : int -> int function
main : int -> int function

(b) Output with Dynamic Scoping

- 1.5
- 2.3
- 3.1
- 4.1
- 5.7

Question 3

(a) Symbol Table with Static Scoping after Specific Lines

After Line 2:
a : int = 53 global

After Line 3:
a : int = 53 global
b : int = 120 global

After Line 4:
a : int = 53 global
b : int = 120 global
c : int = 36 global

After Line 5:
a : int parameter to g

After Line 7:
a : int parameter to g
c : int = b (value of b at the time g is called) local to g

After Line 8:
a : int parameter to g
c : int local to g
b : int = a + c local to g

After line 12 (with function f):
a : int parameter to f

After line 13:
a : int parameter to f
b : int = a + 5 local to f

After line 17 (within function main):
a : int = 53 global
b : int = 120 global

c : int = 36 global
main : int -> int function
f : int -> int function
g : int -> int function

After line 19:
b : int = a local to main (a is 53 globally, so b is now 53 in main)

After line 21:
b : int = a local to main
c : int = b (value of b local to main which is 53) local to main

(b)

- 1.120
- 2.53
- 3.58 (printed from within f, before g is called)
- 4.120 (printed from within g, where b is global b)
- 5.178 (printed from within g, after local b is modified)
- 6.178

Question 4

Line 17:
a: int = 53 global
b: int = 120 global
c: int = 36 global
g: int -> int function
f: int -> int function
main : int function

Line 19:
a: int = 53 global
b: int = 53 local in main :: int = 120 global
c: int = 36 global
g: int -> int function
f: int -> int function
main : int function

Line 12:
a: int = 53 local in f :: int = 53 global
b: int = 53 local in main :: int = 120 global
c: int = 36 global
g: int -> int function

f: int -> int function
main : int function

Line 13:
a: int = 53 local in f :: int = 53 global
b: int = 58 local in f :: int = 53 local in main :: int = 120 global
c: int = 36 global
g: int -> int function
f: int -> int function
main : int function

Line 5:
a: int = 53 local in g :: int = 53 local in f :: int = 53 global
b: int = 58 local in f :: int = 53 local in main :: int = 120 global
c: int = 36 global
g: int -> int function
f: int -> int function
main : int function

Line 7:
a: int = 53 local in g :: int = 53 local in f :: int = 53 global
b: int = 58 local in f :: int = 53 local in main :: int = 120 global
c: int = 120 local in g :: int = 36 global
g: int -> int function
f: int -> int function
main : int function

Line 8:
a: int = 53 local in g :: int = 53 local in f :: int = 53 global
b: int = 173 local in g :: int = 58 local in f :: int = 53 local in main :: int = 120 global
c: int = 120 local in g :: int = 36 global
g: int -> int function
f: int -> int function
main : int function

Line 11:
a: int = 53 local in f :: int = 53 global
b: int = 58 local in f :: int = 53 local in main :: int = 120 global
c: int = 36 global

g: int -> int function
f: int -> int function
main : int function
Line 16:
a: int = 53 global
b: int = 53 local in main :: int = 120 global
c: int = 36 global
g: int -> int function
f: int -> int function
main : int function
Line 21:
a: int = 53 global
b: int = 53 local in main :: int = 120 global
c: int = 111 local in main :: int = 36 global
g: int -> int function
f: int -> int function
main : int function
Line 24:
a: int = 53 global
b: int = 120 global
c: int = 36 global
g: int -> int function
f: int -> int function
main : int function