

Assignment 6
CSCI 3136: Principles of Programming Languages
Due April 8, 2019

Assignments are due on the due date before 23:59. All assignments must be submitted electronically via the course SVN server. Plagiarism in assignment answers will not be tolerated. By submitting your answers to this assignment, you declare that your answers are your original work and that you did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

Submission instructions: Create a directory a6 at the top level of your SVN repository. Place a single PDF file named a6.pdf containing your answers into this directory and submit it via `svn commit`.

Question 1 (10 marks) Consider the following Scheme program:

```
(define x 8)
(define y 3)

(define (f z)
  (display (+ z y)))

(define (g f)
  (let ((y 5))
    (f x)))

(define (h)
  (let ((x 80))
    (g f)))

(h)
```

What does this program output assuming Scheme uses

- (a) Static binding? (Scheme’s actual binding rule)
- (b) Dynamic and deep binding? (hypothetical)
- (c) Dynamic and shallow binding? (hypothetical)

Question 2 (5 marks) In class, we discussed how to implement static scoping using a static chain. We discussed how to use the static chain to access a variable we want to update at runtime. What we did not discuss is how to maintain this static chain when we make recursive calls. Your task in this question is to provide the details of computing the static link for a function that is being called, that is, how to find the stack frame the static link of the newly created stack frame needs to point to. (The compiler should be able to determine at compile time how many jumps up the caller’s static chain are necessary to reach the stack frame that is the target of the static link in the callee’s stack frame.)

Question 3 (5 marks) Binary arithmetic operations are left-associative in most programming languages (because most of them are in mathematics). On the other hand, for efficiency reasons, many languages allow the compiler to evaluate the operands of any binary operator in any order. Are these two facts contradictory? Why or why not?