

SI 413 Fall 2021: Homework 9

Due Monday, November 8

Your name:

Citations and collaborators:

Comments, suggestions, or questions for your instructor:

Fill out the first row of the table on a 0-5 scale before turning in.

This rubric is also available on the website under “Admin”:

- **5:** Solution is completely correct, concisely presented, and neatly written.
- **4:** The solution is mostly correct, but one or two minor details were missed, or the presentation could be better.
- **3:** The main idea is correct, but there are some significant mistakes. The presentation is somewhat sloppy or confused.
- **2:** A complete effort was made, but the result is mostly incorrect.
- **1:** The beginning of an attempt was made, but the work is clearly incomplete.
- **0:** Not submitted.

Problem	1	2	3	4	Total
Self-assessment					
Final assessment					

1 Pass by what?

Here is a small SPL program with a single function that gets called twice.

```
new a := 20;
new b := 10

new foo := lambda x {
  x := x + x;
  ret := x * b;
}

write foo@a;
write a;
write foo@b;
write b;
```

Clearly four numbers will get printed by this piece of code. Tell me what those four numbers will be under:

- a) Pass by value
- b) Pass by reference
- c) Pass by value/result

2 Value/result example

In Java function parameters are passed *by sharing* by default. In class, we saw a different kind of parameter passing mode called pass by value/result. Write a small Java program demonstrating that function parameters in Java really are passed by sharing and not by value/result. That is, your program should do something different in each of these parameter passing modes.

As always, I want you to come up with your own examples! You can work together on homeworks, but the examples you turn in should be unique.

Clearly specify what the result of your function would be under pass by sharing and under pass by value/result.

4 Barbarian Sausage Gang

Consider the following Scheme program:

```
(define (f x y)
  (display x)
  (display 'a)
  (display y)
  x)

(define (g x y)
  (cond ((eqv? x 'g)
         (display x))
        ((eqv? x y)
         (display 'e))))

(g (f ?? ??) (f ?? ??))
```

- Fill in the ?? blanks in the function call so that the program prints **barbarian** under call-by-name evaluation rules.
- Fill in the ?? blanks in the function call so that the program prints **sausage** under applicative-order evaluation rules.
- Fill in the ?? blanks in the function call so that the program prints **gang** under normal-order evaluation rules.