

Worksheet 7A

Quiz 1: Semantics: Functions as Values

What should the following programs *evaluate* to?

Program

```
(fun (f it)
  (it 5))

(fun (inc x)
  (+ x 1))

(f inc)
```

Result

Quiz 2: Evaluator

Fill in the blanks to extend `eval` to handle function definitions and calls.

```
fn eval(&self, e: &Expr, env: &mut Env) -> Val {
  match e {
    ...
    Expr::Defn(defn) =>
      _____,
    Expr::Call1(f, e1) => {
      let v1 = eval(e1, env)?;
      _____
      _____
      _____
      _____
    }
    ...
  }
}
```

Quiz 3: Compiler: Fun Call

Fill in the blanks to compile function calls.

```
Call1(f, e1) =>
```

Quiz 4: Semantics: Anonymous Functions

What should the following program *evaluate* to?

Program

```
; anon function that takes `it`  
; and returns `it 5`  
(let (f (fn (it) (it 5)))  
  
; anon function that takes `z`  
; and returns `( + z 1 )`  
(let (inc (fn (z) (+ z 1)))  
  
  (f inc)  
  
))
```

Result

Quiz 5: Semantics: Arity

What should the following program *evaluate* to?

Program

```
(let (f (fn (it) (it 5)))  
  
(let (add (fn (x1 x2) (+ x1 x2)))  
  
  (f add)  
  
))
```

Result

Quiz 6: Your turn!

What is something you found confusing in today's lecture (or earlier)?