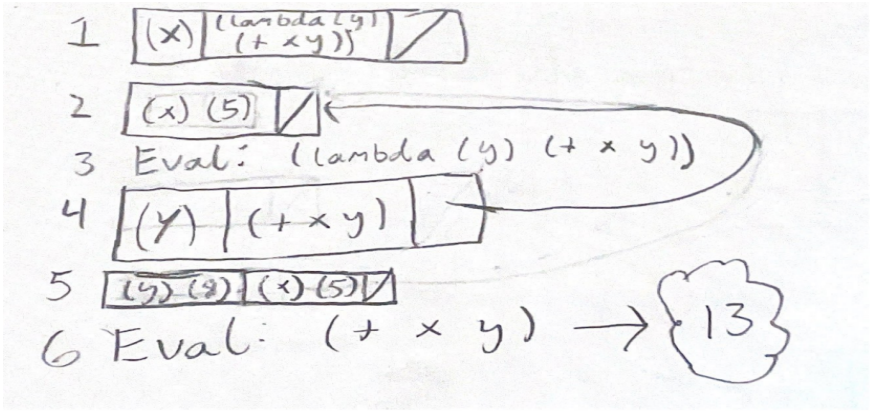
CSSE304 Exam 2 Paper Review

Instructions

Please complete the following problems. If you have questions, come up and ask. I will send out an answer key after the review session is over.

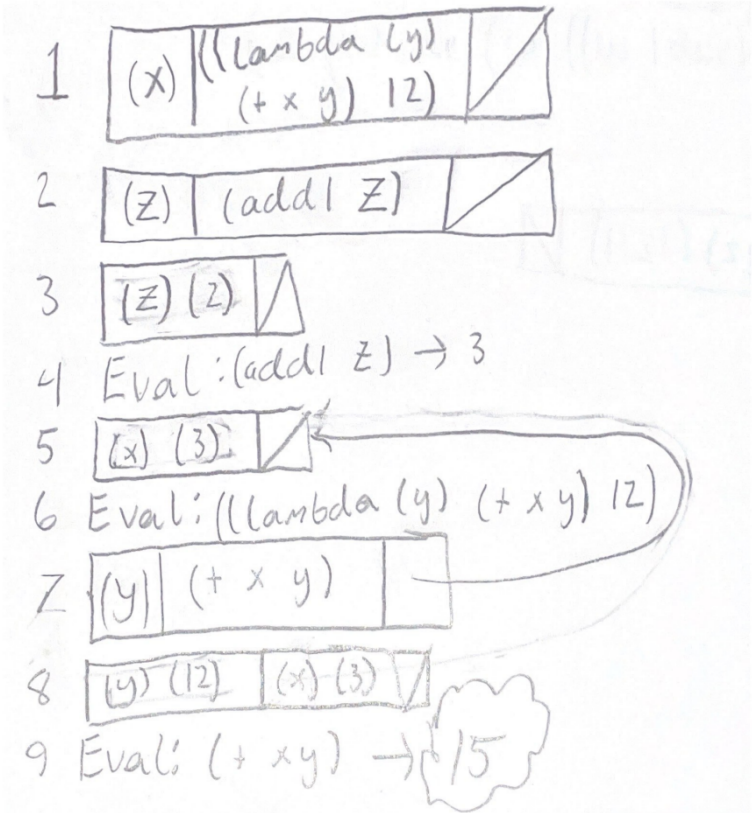
1. **Problem 1:** Draw an E&C diagram for the following code

   ```
   (((lambda (x) (lambda (y) (+ x y))) 5) 8)
   ```



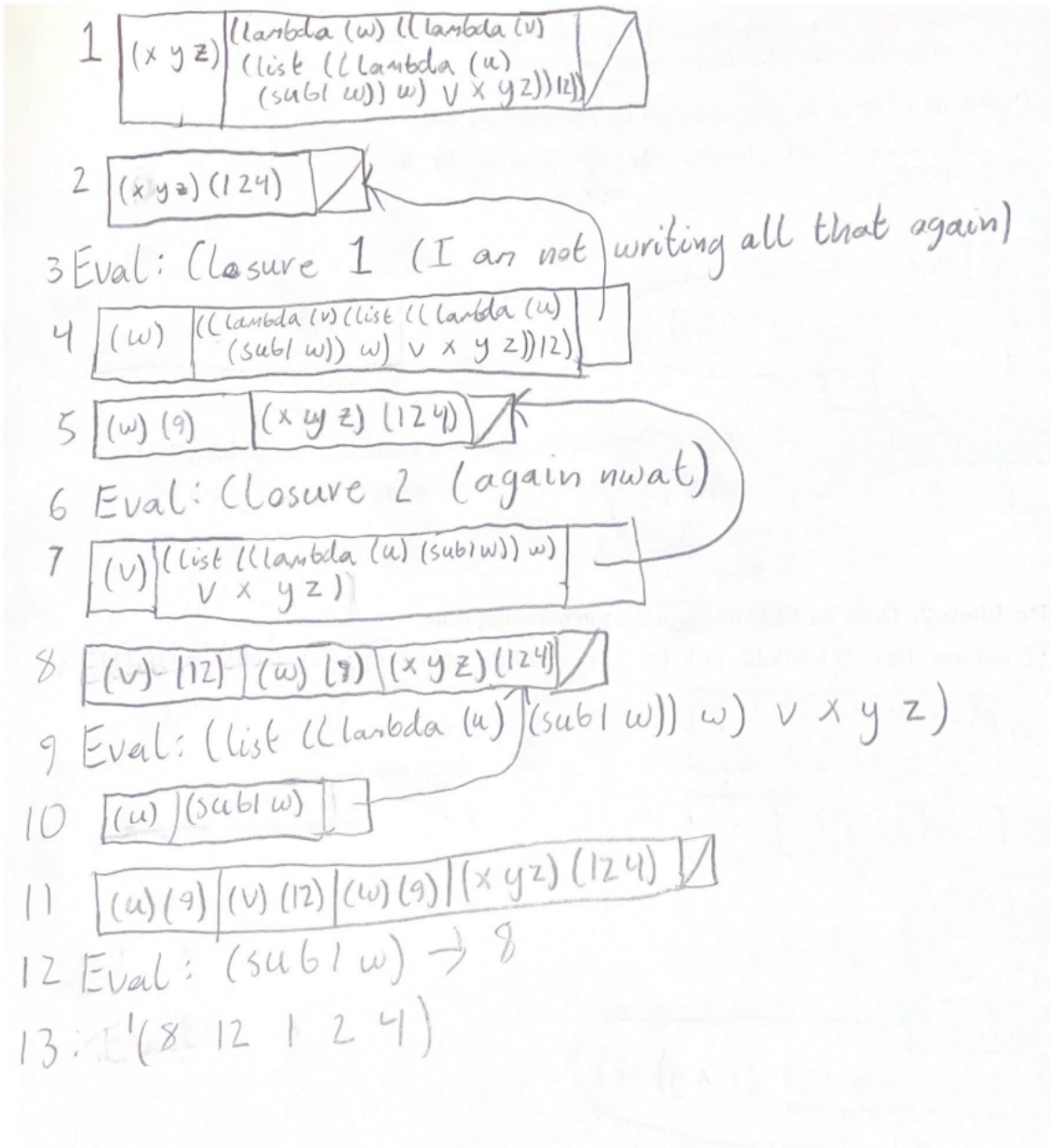
2. **Problem 2:** Draw an E&C diagram for the following code

   ```
   ((lambda (x) ((lambda (y) (+ x y)) 12)) ((lambda (z) (add1 z)) 2))
   ```



3. **Problem 3:** Draw an E&C diagram for the following code

```
(((lambda (x y z) (lambda (w) ((lambda (v) (list ((lambda (u)
    (sub1 w)) w) v x y z)) 12))) 1 2 4) 9)
```



4. **Problem 4:** What does an abstract datatype consist of?

- An abstract data type consists of:
  - Interface (how the user sees it)
  - An indication of the runtime of operations
- A Data structure additionally:
  - The underlying representation
  - The implementation of the functionality

5. **Problem 5:** Implement a binary tree datatype based on the following grammar.

   <bintree> ::= <number> | (<number> <bintree> <bintree>)

```
(define-datatype bintree bintree?
  [leaf-node
    (datum number?)]
  [interior-node
    (datum number?)
    (left-tree bintree?)
    (right-tree bintree?)])
```

6. **Problem 6:** Provide an implementation for snlist-recur

```
(define snlist-recur
  (lambda (flist fatom init)
    (letrec ([helper (lambda (l)
      (cond [(null? l) init]
     [(or (pair? (car l))
                          (null? (car l))) (flist (helper (car l))
     (helper (cdr l)))]
     [else (fatom (car l) (helper (cdr l)))]))])
      helper)))
```

7. **Problem 7:** Provide an implementation for Y-combinator *(may be smart to look at other things implemented in class)*

```
(define foo
  (lambda (g)
    (lambda (n)
      (if (zero? n)
          1
          (* n (g (- n 1)))))))

(define Y
  (lambda (f)
    ((lambda (x)
       (f (lambda (t)
            ((x x) t))))
     (lambda (x)
       (f (lambda (t)
            ((x x) t)))))))
```