

## CS424/524-01: Programming Languages

## Extra Credit

Due: See Canvas for Due Dates

(up to 5 points added to a test grade)

These problems cover material from Chapter 15 of the textbook.

Chapter 15: Work the following problems. Point values are provided.

1. (1) What is the result of each of the following statements [In the event a list is an answer, just write the list form not the annotated version - Example: (a b c) **NOT** (list a b c)]

```
(+ (car '(1 2 3)) (caar '((1) 10 30)))
```

- Add
- Car takes first element of list: 1
- Caar takes first element of first element: 1
- **Result is 2**

2. (1) What is wrong with the following expression. Do not just give me an error message. Explain it.

```
(caar '(a (b c d)))
```

- Caar takes first element of first element but the first element is 'a' which is a symbol. Therefore it's an atomic value

3. (1) Write a Scheme/Lisp expression to pull the value *pear* out of the following list use only car and cdr, no other combinations.

```
'(orange (apple grape ((pear) raisin)) lemon)
```

- Cdr '(orange (apple grape ((pear) raisin)) lemon) – return everything but 1<sup>st</sup> element
  - ((apple grape ((pear) raisin)) lemon)
- Car ((apple grape ((pear) raisin)) lemon) – return first element
  - (Apple grape ((pear) raisin))
- Cdr (Apple grape ((pear) raisin))
  - (Grape ((pear) raisin))
- Cdr (grape ((pear) raisin))
  - ((pear) raisin)
- Car ((pear) raisin)
  - (Pear)
- Car (pear)
  - Pear
- Answer: (car (car (cdr (cdr (car (cdr '(orange (apple grape ((pear) raisin)) lemon)))))))

4. (1) Write a single lisp/scheme expression to insert the new value 'z into the beginning of the list '(a b) to produce '(z a b)

- (Cons 'z '(a b))

5. (1) Explain what the following Scheme/LISP function (named EXF1) does. In other words, tell me what it accomplishes, not just describe the step-by-step logic:

```
(define (EXF1 S L)
  (cond ((null? L) '())
        ((equal? S (car L)) L)
        (else (EXF1 S (cdr L))))
)
```

- Takes two arguments of S and L and checks if L is empty. If empty it returns an empty list. Checks if first element of L is equal to S, if not it calls the function again with the rest of the list
- EXF1 uses recursion to search in some list L for a target S