Lisp Text Questions:

2.2:

The well formed lists are:
(A B (C))
((A) (B))
(A (B (C)))
(((A) (B)) (C))

2.4

((BOWS ARROWS) (FLOWERS CHOCOLATES))

2.6

() → NIL
(()) → (NIL)
((())) → ((NIL))
(() ()) → (NIL NIL)
(() (())) → (NIL (NIL))

2.13

FUN → CAAAR
IN → CAADR
THE → CADADR
SUN → CAADDR

2.15

CAR → (A B)
CDDR → ((E F))
CADR → (C D)
CDAR → (B)
CADAR → B
CDDAR → NIL
CAAR → A
CDADDR → NIL
CADDDR → F

2.16

CAR cannot be applied to a single element and after the first CAR there is just FRED. It would result in an error. CAAR cannot be performed on (FRED NIL)

Sebastia Chapter 1 Review Questions

6
Most of UNIX is written in C.

7
The disadvantage of having too many features in a language is increased complexity, which can lead to harder learning, maintenance, and potential for errors.

8
User-defined operator overloading can harm readability by making it unclear what operations do without knowing the specific definitions.

9
In C, arrays and pointers are not orthogonal—arrays decay into pointers, leading to inconsistent behavior.

10
ALGOL 68 used orthogonality as a primary design criterion.

11
The goto statement is used to build more complicated control statements in languages that lack them.

12
A program is reliable if it performs correctly under all conditions and handles errors gracefully.

13
Type checking parameters ensures correctness, prevents errors, and improves reliability.

14
Aliasing occurs when two or more names refer to the same memory location.

15
Exception handling is a mechanism to manage runtime errors or unexpected events.

16
Readability is important to writability because clear, understandable code is easier to write and modify.

20
The two deficiencies discovered were lack of modularity and poor control of abstraction.

21
The three fundamental features of an object-oriented programming language are encapsulation, inheritance, and polymorphism.

22
Simula 67 was the first language to support these three features.

23
Readability and execution efficiency are often in direct conflict.

24
The three general methods are compilation, pure interpretation, and hybrid implementation.

25
A compiler produces faster program execution than a pure interpreter.

29
- Easier debugging
- Portability
- No compilation
- Dynamic flexibility