CHA VUE
1/26/25

HOMEWORK 1

Lisp text Questions

2.2. ((A) (B)), (A (B (C))) and (((A) (B)) (C)) have properly balanced parentheses.

2.4. ((BOWS ARROWS) (FLOWERS CHOCOLATES))

2.6. ( ) = NIL; (( )) = (NIL); ((( ))) = ((NIL)); (( ) ( )) = (NIL NIL); (( ) (( ))) = (NIL (NIL))

2.13.

Start	(((FUN)) (IN THE) (SUN))
C...AR	(FUN)
CADR	FUN
Start	(((FUN)) (IN THE) (SUN))
C...DR	((IN THE) (SUN))
C..ADR	(IN THE)
CAADR	IN
Start	(((FUN)) (IN THE) (SUN))
C...DR	((IN THE) (SUN))
C..ADR	(IN THE)
CDADR	THE
Start	(((FUN)) (IN THE) (SUN))
C...DDR	(SUN)
CADDR	SUN

2.15.

Function	**Result**
CAR	(A B)
CDDR	((E F))
CADR	(C D)
CDAR	(B)
CDDAR	B
CDDAR	B
CDADDR	A
CDADDR	A
CADADR	F

2.16. This would give an ERROR because FRED is not a list.

Sebesta Chapter 1 review questions

6. Most of Unix is written in C.

7. A language that has too many features can be confusing and fall behind in efficiency. This is because programmers who know these features can use them and those who do not won't be able to maximize their tools.

8. This can affect readability if the programmer does not do it sensibly. Errors in programs can go undetected and can lead to code absurdities that cannot be discovered by the compiler.

9. A member of a structure can be any data type except void or a structure of the same type.

10. ALGOL 68

11. A set of primitive constructs can be combined to build control and data structures of the language.

12. If a program is reliable, it performs to its specifications under all conditions.

13. This is because the earlier an error is detected the less expensive it is to fix it and make repairs.

14. Aliasing is having two or more distinct names in a program that can access the same memory cell.

15. The ability of a program to intercept run-time errors, take corrective measures, and then continue.

16. Because programs that are difficult to read are difficult to write.

20. Incompleteness of type checking and inadequacy of control statements.

21. Data abstraction, inheritance, and dynamic method binding.

22. Simula 67

23. Reliability and cost of execution.

24. Compiler implementation, Pure interpretation, and Hybrid implementation.

25. A compiler is about 10 – 100 times faster than pure interpretation.

29. It allows easy implementation of many source-level debugging operations.