Questions:

1. From Lisp text: Questions 2.2, 2.4 (page 34), 2.6 (page 38), 2.13. 2.15, 2.16 (Page 49)

2.2:

(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 – (F)

CADADDR – F

2.16:

Error. CAR of (FRED NIL) = FRED, CAR of FRED = Error.

2. From Sebesta Chapter 1 review questions: Questions 6 through 16, 20 through 25, and 29.

6. Most of UNIX is written in C.

7. Having too many features can make the language too complex and harder to learn, it could lead to misuse of features or disuse, and it increases the risk of inconsistent coding styles.

8. If operator overloading is done in a non-intuitive or inconsistent way, it can confuse readers about the behavior of the overloaded operators, making the program harder to understand.

9. In C, arrays cannot be returned from functions, while structs can. This inconsistency is an example of a lack of orthogonality.

10. ALGOL 68 used orthogonality as a primary design criterion.

11. The GOTO statement is used as a primitive control statement to build more complicated control statements in languages that lack them.

12. A program is reliable if it performs according to its specifications under all conditions, that is including handling unexpected inputs or errors gracefully.

13. Type checking ensures that the arguments passed to a subprogram match the expected types, preventing runtime errors and improving the reliability of the program.

14. Aliasing occurs when two or more different names (or references) are used to access the same memory location. It can lead to unintended side effects, such as modifying the memory using one alias also affects the other.

15. Exception handling is a programming mechanism that allows a program to detect, handle, and recover from unexpected errors or unusual conditions during execution without crashing. It typically involves try-catch (or similar) constructs to manage errors gracefully.

16. Readability is essential to writability because writing code often involves reading and understanding existing code. Clear and readable code makes it easier to reason about the program's logic and reduces the likelihood of errors during modification or extension.

20. Incompleteness of type checking: This led to increased bugs and runtime errors.
    Inadequacy of control statements: The excessive use of goto statements made programs difficult to understand and maintain.

21. Encapsulation: Bundling of data and methods into objects.

    Inheritance: Mechanism for creating a new class based on an existing class.

    Dynamic method binding (polymorphism): Determining which method to invoke at runtime.

22. Smalltalk.

23. Readability vs. writability: Adding features for writability, like operator overloading, can make a language harder to read.

Flexibility vs. reliability: Allowing more flexibility, for example dynamic typing, can lead to less reliable programs due to increased runtime errors.

24. Compilation: Translating the source code into machine code.

    Pure interpretation: Executing source code directly line by line.

    Hybrid implementation: Combining compilation and interpretation (e.g., Java's bytecode and JVM).

25. A compiler produces faster program execution because it translates the code into machine code beforehand, eliminating runtime translation.

29. Improved error diagnosis and debugging: Pure interpreters can immediately identify and report errors during execution, making it easier to debug and correct programs.

    Platform independence: The source code can run on any platform that has the interpreter implemented, eliminating the need for recompilation.

    Dynamic execution: Pure interpreters allow for the execution of dynamic code (e.g., code generated or modified at runtime), which is not possible with precompiled languages.

    Immediate execution: Programs can be executed directly without needing a compilation step, which is especially useful during development for quick testing of code.