Joe Whann

CSCI 330

Homework 1 Q1&2

Q1

2.2

(A B (C)

- Not balanced

((A) (B)):

- Well-formed. The parentheses are balanced and properly nested.

A B) (C D:

- Not well-formed. Closing parenthesis) appears before an opening parenthesis (.

(A (B (C))):

- Well-formed. The parentheses are balanced and nested correctly.

(A (B (C)):

- Not well-formed. Missing a closing parenthesis) at the end.

(((A) (B)) (C)):

- Well-formed. The parentheses are balanced and properly nested.

2.4

Final Parentheses notation is ((BOWS ARROWS) (FLOWERS CHOCOLATES))

2.6

Final Matches:

1. () ↔ NIL
2. (()) ↔ (NIL)
3. ((())) ↔ ((NIL))

4. (() ()) ↔ (NIL NIL)
5. (() ((()))) ↔ (NIL (NIL))

2.13

Step	Result
Start	(((FUN)) (IN THE) (SUN))
CAR	((FUN))
CAR	(FUN)
CAR	FUN
Start	(((FUN)) (IN THE) (SUN))
CDR	((IN THE) (SUN))
CAR	(IN THE)
Start	(((FUN)) (IN THE) (SUN))
CDR	((IN THE) (SUN))
CDR	(SUN)
CAR	SUN

2.15

Final Table

Function	Result
CAR	(A B)
CDR	((C D) (E F))
CADR	(C D)
CDAR	(B)
CDDR	((E F))

Function Result

CDDDAR NIL

CADDDR E

2.16

CAAR is equivalent to (CAR (CAR …)):

1. Apply the first CAR: (CAR (FRED NIL)) → FRED.
2. Apply the second CAR: (CAR FRED).

Since CAR cannot operate on FRED (a symbol, not a list), this will result in an **error**.

Q2

Sebesta questions

6. In what language is most of UNIX written?

- **Answer**: C.

7. What is the disadvantage of having too many features in a language?

- **Answer**: Too many features make a language more complex, leading to difficulty in learning, understanding, and maintaining code.

8. How can user-defined operator overloading harm the readability of a program?

- **Answer**: Poorly designed operator overloading can obscure the intended meaning of operations, making the program harder to understand.

9. What is one example of a lack of orthogonality in the design of C?

- **Answer**: Arrays in C are not orthogonal. For example, an array name behaves like a pointer in some contexts but not in others (e.g., sizeof).

10. What language used orthogonality as a primary design criterion?

- **Answer**: ALGOL 68.

11. What primitive control statement is used to build more complicated control statements in languages that lack them?

- **Answer**: Goto.

12. What does it mean for a program to be reliable?

- **Answer**: A reliable program performs its specified functions under all conditions without failure.

13. Why is type checking the parameters of a subprogram important?

- **Answer**: Type checking prevents type mismatches, reducing runtime errors and improving program reliability.

14. What is aliasing?

- **Answer**: Aliasing occurs when two or more variables reference the same memory location, which can cause unexpected behavior or bugs.

15. What is exception handling?

- **Answer**: Exception handling is a mechanism for managing runtime errors or unusual conditions by detecting, signaling, and responding to them.

16. Why is readability important to writability?

- **Answer**: Readable code is easier to understand, which makes it simpler to modify, extend, or debug, thereby improving writability.

20. What two programming language deficiencies were discovered as a result of the research in software development in the 1970s?

- **Answer**: Inadequate support for data abstraction and insufficient modularity.

21. What are the three fundamental features of an object-oriented programming language?

- **Answer**: Encapsulation, inheritance, and polymorphism.

22. What language was the first to support the three fundamental features of object-oriented programming?

- **Answer**: Smalltalk.

23. What is an example of two language design criteria that are in direct conflict with each other?

- **Answer**: Readability and writability often conflict, as making a language easier to write may sacrifice clarity and understanding.

24. What are the three general methods of implementing a programming language?

- **Answer**: Compilation, pure interpretation, and hybrid implementation.

25. Which produces faster program execution, a compiler or a pure interpreter?

- **Answer**: A compiler.

29. What are the advantages in implementing a language with a pure interpreter?

- **Answer**: Pure interpreters provide dynamic error checking, ease of debugging, and allow immediate execution of code without the need for a separate compilation step.