

2.2. Which of these are well-formed lists? That is, which ones have properly balanced parentheses?

(A B (C) **not balanced**

((A) (B)) **balanced**

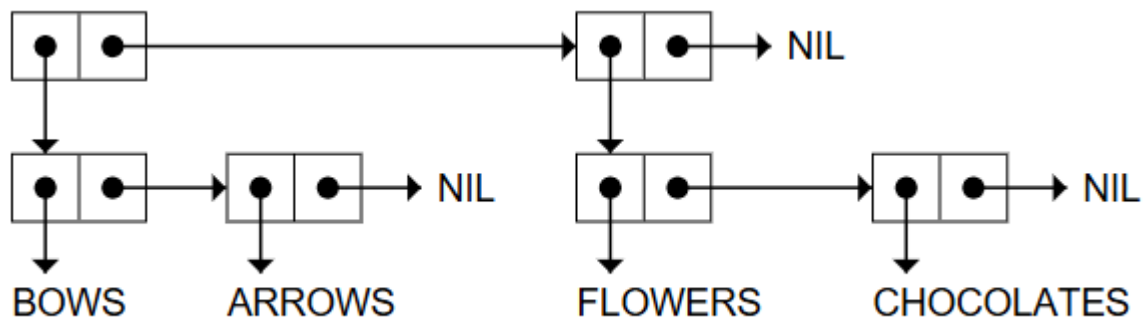
A B)(C D) **not balanced**

(A (B (C)) **not balanced**

(A (B (C))) **balanced**

((A) (B)) (C)) **balanced**

2.4 What is the parenthesis notation for this cons cell structure?



((BOWS ARROWS) (FLOWERS CHOCOLATES))

2.6 Match each list on the left with a corresponding list on the right by substituting NIL for () wherever possible. Pay careful attention to levels of parenthesization.

()	((NIL))
(())	NIL
((()))	(NIL)
(() ())	(NIL (NIL))
(() (()))	(NIL NIL)

() = NIL

(()) = (NIL)

((())) = ((NIL))

(() ()) = (NIL NIL)

(() (())) = (NIL (NIL))

2.13. Write down tables similar to the one above to illustrate how to get to each word in the list (((FUN)) (IN THE) (SUN)).

Step:	Result
CAAAR	FUN
CAADR	IN
CADADR	THE
CADDR	SUN

2.15 Using the list ((A B) (C D) (E F)), fill in the missing parts of this table.

Function	Result
CAR	(A B)
CDDR	(E F)

CADR	(C D)
CDAR	(B)
CADR	(B)
B	CADAR
CDDAR	NIL
A	CAAR
CDADDR	(F)
CADADDR	F

2.16 What does CAAR do when given the input (FRED NIL)?

CAAR \rightarrow CAR (FRED NIL) = FRED CAR (FRED) = error, not a list.

Questions 6-16:

6. In what language is most of UNIX written?

C

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

It tends to lead to knowing only a subset.

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

It can lead to reduced readability if users are allowed to create their own overloading and do not do it sensibly.

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

C has two structured data types (arrays and structs). Arrays can be returned from functions, but structs can't.

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

ALGOL 68 used orthogonality as a primary design criterion.

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

GOTO selection statement.

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

A program is said to be reliable if it performs to its specifications under all conditions.

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

Because it can lead to lots of hard-to-debug errors.

14. What is aliasing?

Two or more distinct names that can be used to access the same memory cell.

15. What is exception handling?

It is the process of responding to occurrences during computation, often changing the normal flow of program execution, and is provided by specialized programming language constructs or computer hardware mechanisms.

16. Why is readability important to writability?

Readability is important to writability because if a programming language is difficult to read/understand, then it can make it difficult for a programmer to create new code for others.

Questions 20-25:

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

Incompleteness of type checking and inadequacy of control statements.

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

Data abstraction, inheritance, and dynamic method dispatch.

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

Smalltalk

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

Reliability / Cost of execution

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

Compilation, pure interpretation, hybrid implementation systems.

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

Compiler

Question 29:

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

The advantage is that a pure interpreter allows for the implementation of many source-level debugging operations. Many run-time error messages refer to source-level units.