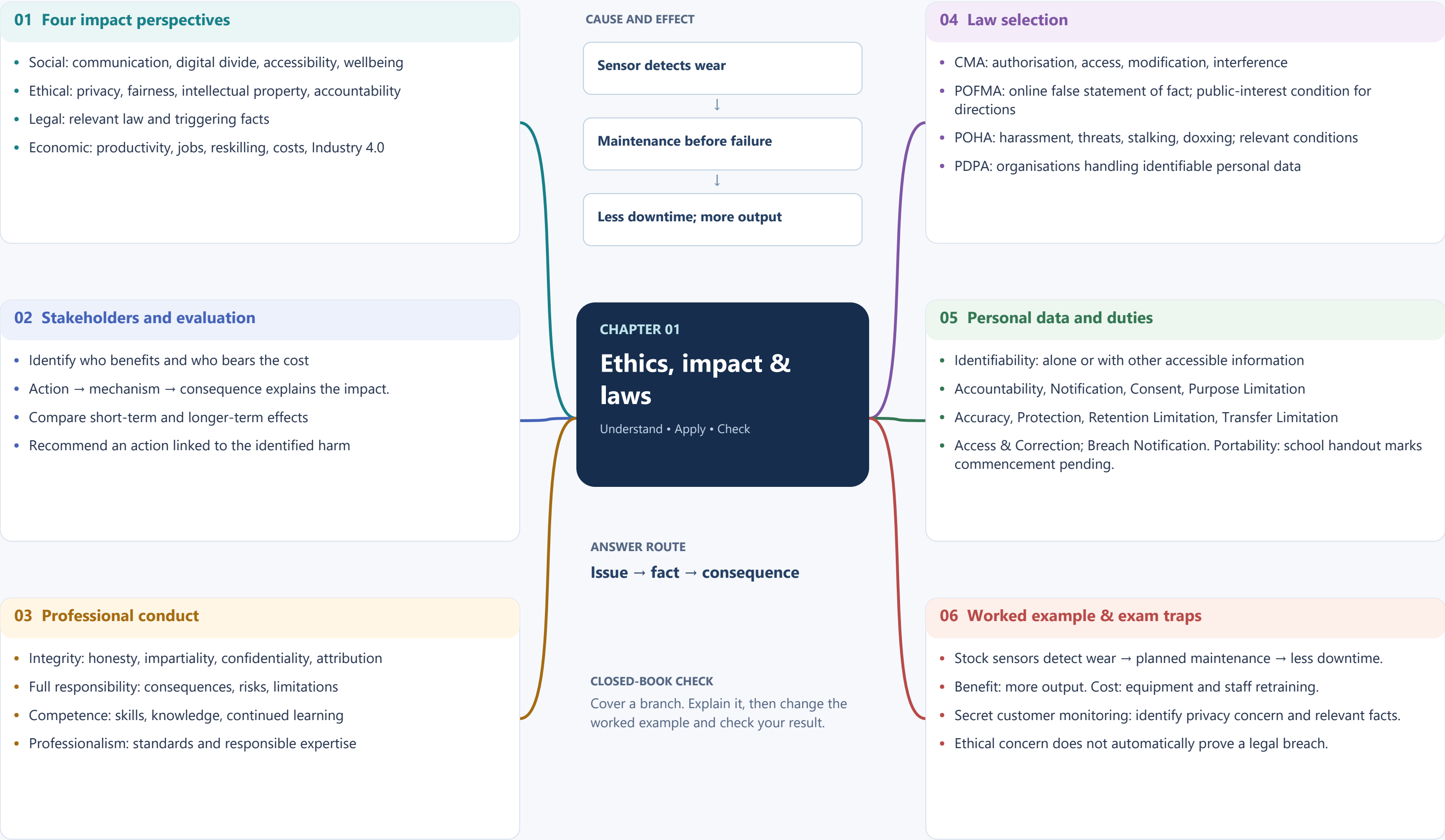
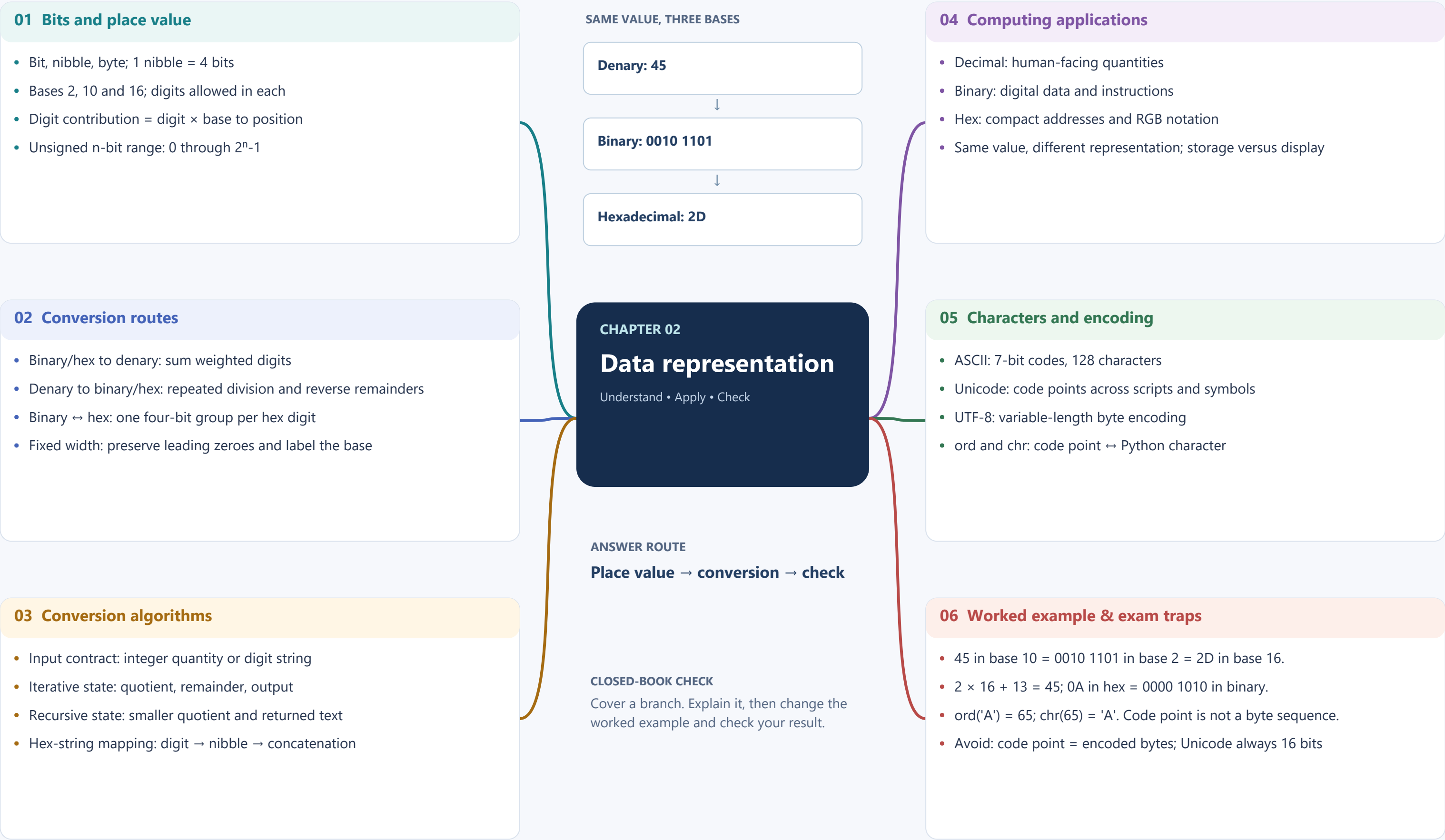


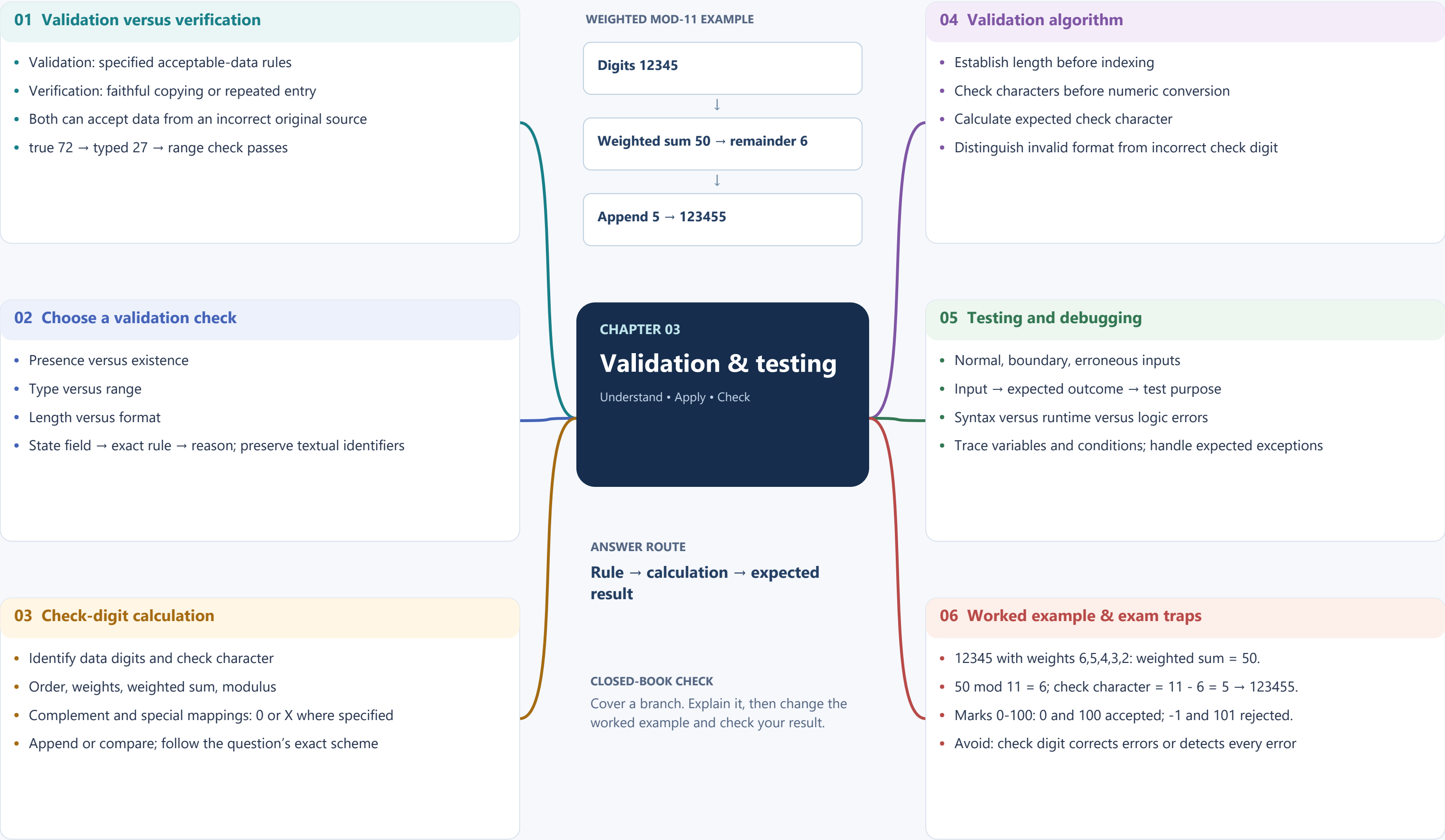
Ethics, impact and laws



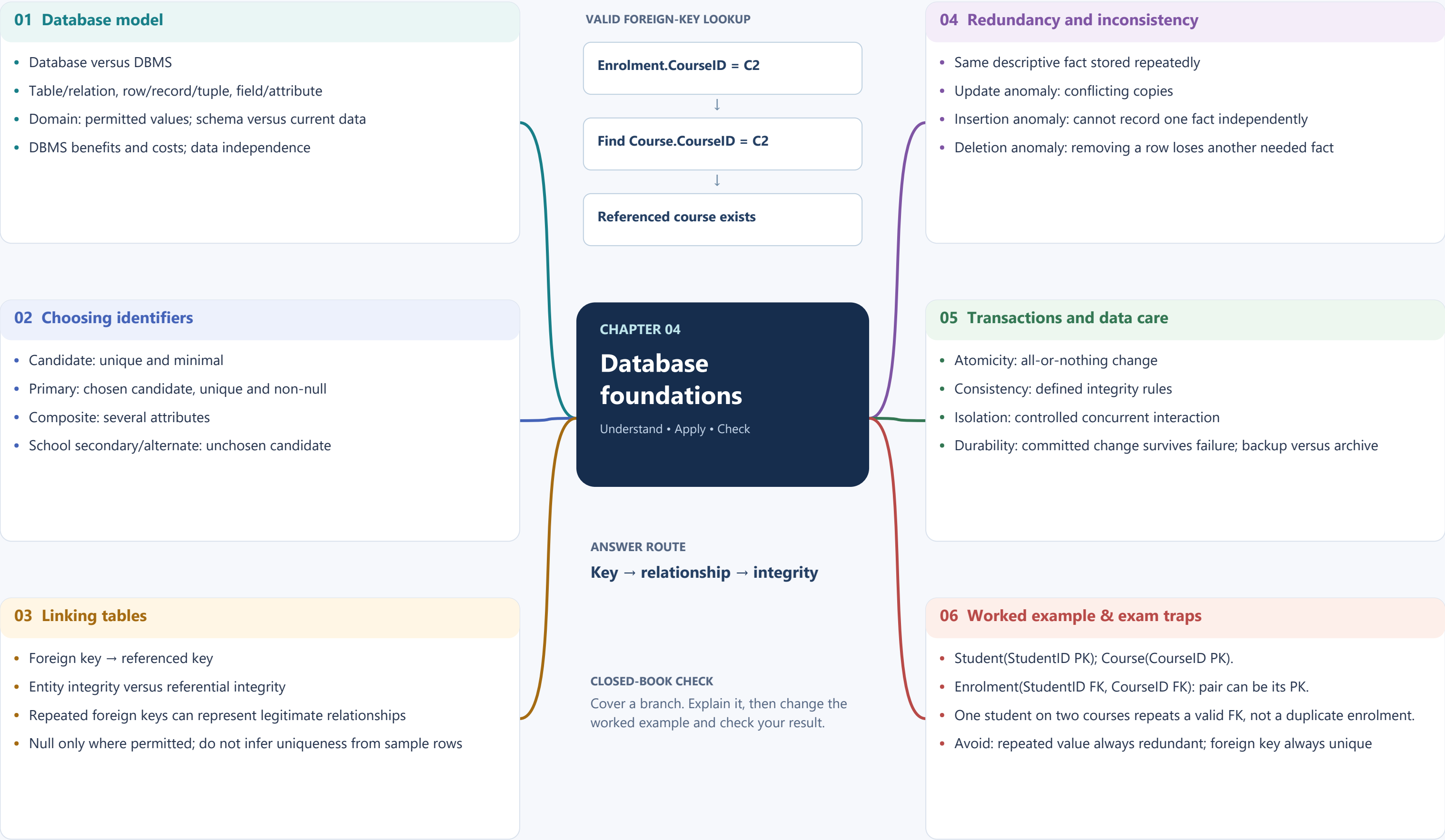
Data representation and character encoding



Validation, verification, check digits and testing



Database foundations, keys and integrity



Normalisation and ER diagrams

01 Read the scenario

- What does one row represent?
- Entity facts versus relationship facts
- Business rules establish keys and cardinalities
- Keep every required attribute through decomposition

02 Functional dependencies

- Determinant → dependent
- A determines B does not mean B determines A
- Partial: proper part of composite key determines non-key fact
- Transitive: key → intermediate non-key fact → another non-key fact

03 Normal-form progression

- UNF: repeating group
- 1NF: atomic values, identifiable rows
- 2NF: remove partial non-key dependencies
- 3NF: remove relevant transitive non-key dependencies

REMOVE A TRANSITIVE FACT

OrderID → CustomerID

↓

CustomerID → CustomerName

↓

Store name in Customer

CHAPTER 05

Normalisation & ERD

Understand • Apply • Check

ANSWER ROUTE

Dependency → decomposition → ERD

CLOSED-BOOK CHECK

Cover a branch. Explain it, then change the worked example and check your result.

04 ER relationships

- Read both directions
- 1:1, 1:M, M:N; cardinality versus optionality
- Foreign key on many side for 1:M
- Resolve M:N with linking entity and appropriate key

05 Worked decomposition

- Order header plus repeating products
- Flatten one row per order-product pair
- Separate Orders, Product and OrderLine
- Separate Customer; Quantity remains on OrderLine

06 Worked example & exam traps

- OrderLine key: (OrderID, ProductID); Quantity depends on the pair.
- ProductID → ProductName is partial: put it in Product.
- OrderID → CustomerID → CustomerName: separate Customer.
- Avoid: smaller tables automatically mean 3NF; names automatically identify

SQL and working with SQLite

01 Decode the requested result

- One row represents what?
- Required columns and source tables
- Conditions, grouping and order
- Order count versus total quantity; preserve meaningful duplicates

02 Read and filter

- SELECT, FROM, aliases
- WHERE: comparisons, Boolean conditions, BETWEEN, IN, LIKE
- NULL: IS NULL and IS NOT NULL
- ORDER BY direction and tie-breaker where required

03 Join and aggregate

- Match the intended key pairs
- INNER JOIN versus LEFT JOIN
- COUNT(*), COUNT(column), SUM, AVG, MIN, MAX
- GROUP BY forms groups; HAVING filters groups

LEFT JOIN AND CHILD COUNT

Customers A, B; orders A, A

↓

Join while retaining B

↓

COUNT(OrderID): A=2, B=0

CHAPTER 06

SQL & SQLite

Understand • Apply • Check

ANSWER ROUTE

Rows → joins → groups → result

CLOSED-BOOK CHECK

Cover a branch. Explain it, then change the worked example and check your result.

04 Define and change data

- CREATE: types, primary/foreign keys, constraints
- INSERT: columns and values
- UPDATE and DELETE: identify affected rows with WHERE
- DROP removes a table; foreign-key enforcement on the connection

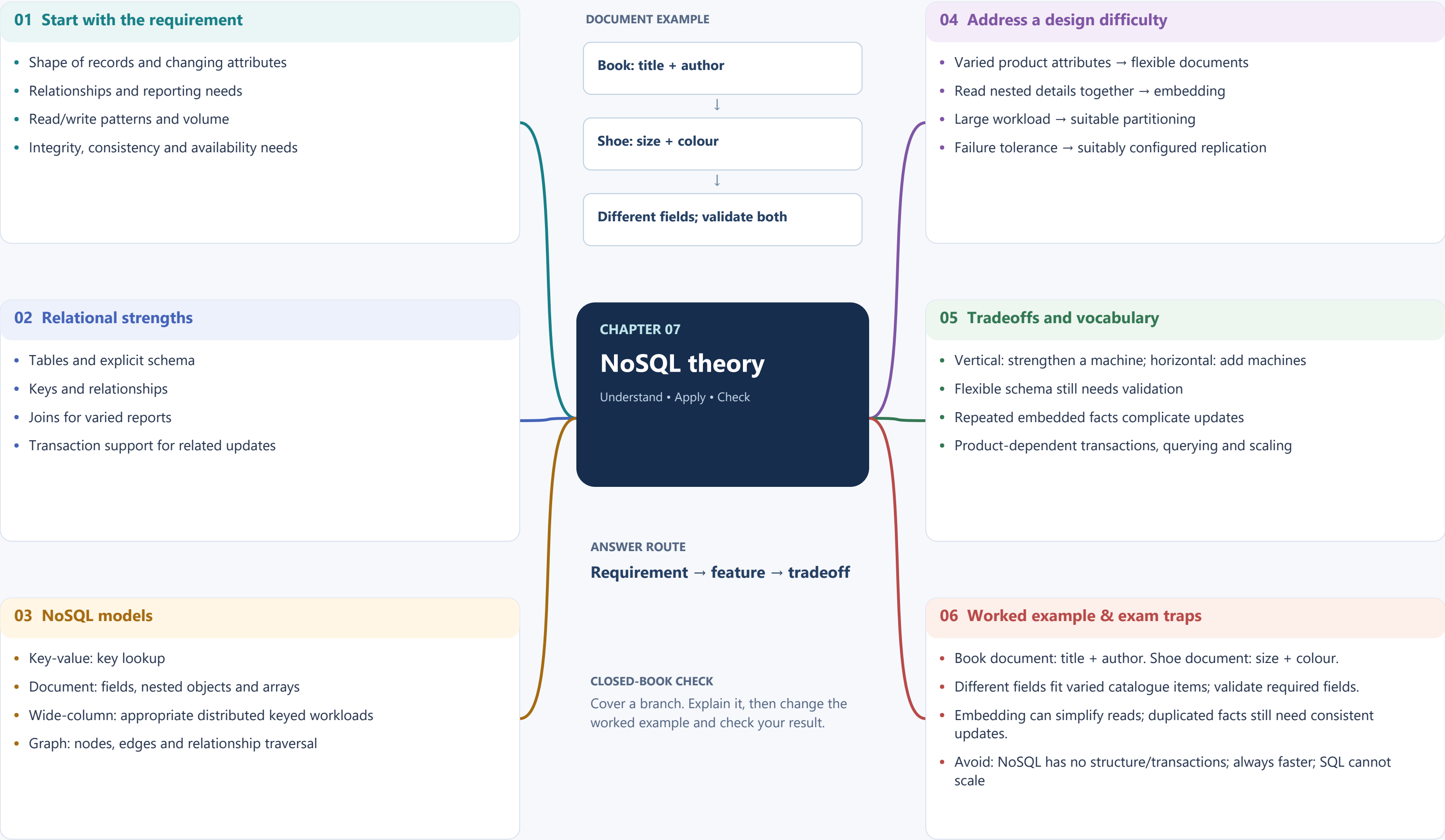
05 Python and SQLite

- Connect → parameterised execute
- Fetch one/all rows; no match behaviour
- Commit changes where needed; close connection
- Placeholders bind values; one-item tuple needs a comma

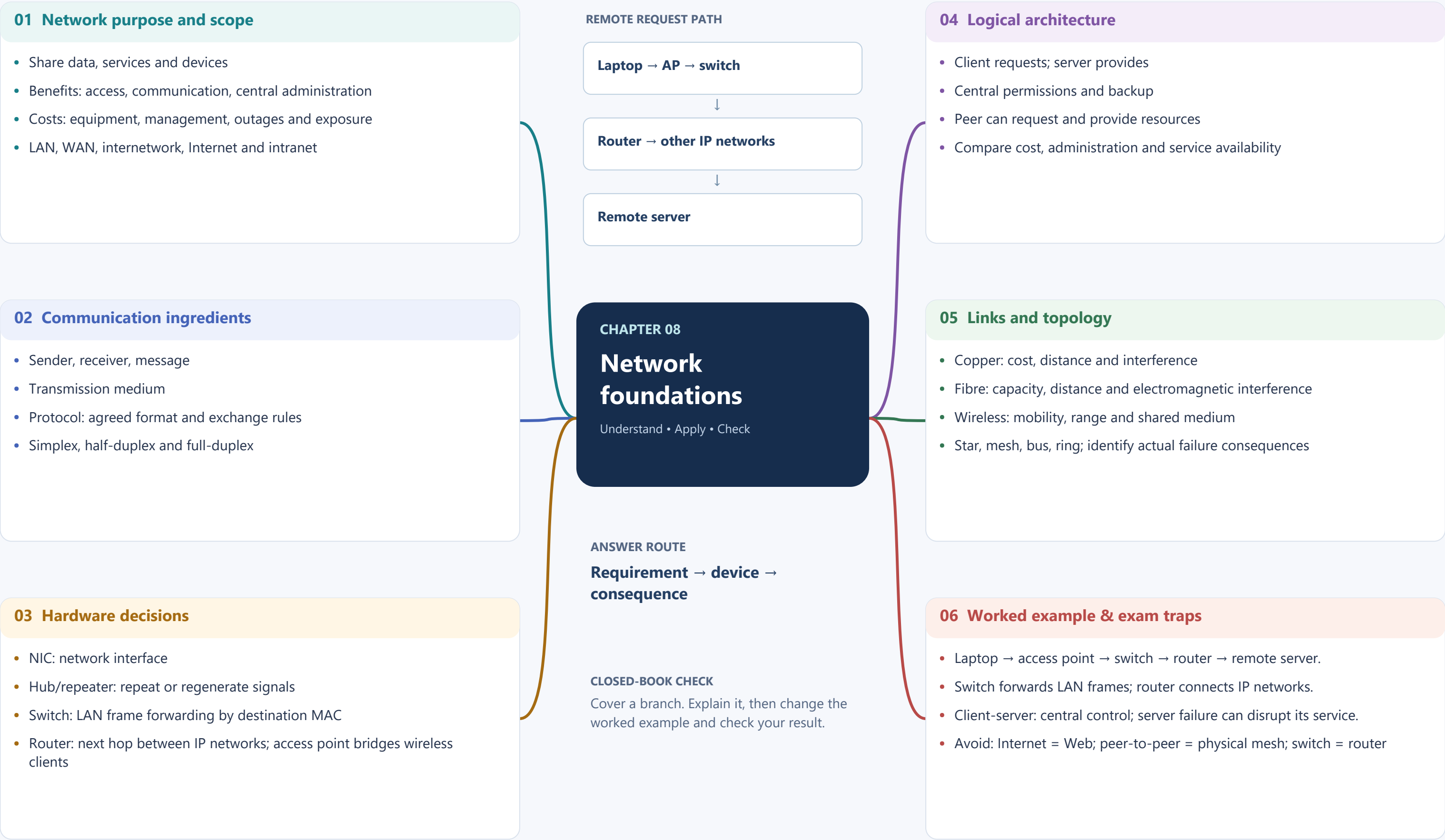
06 Worked example & exam traps

- Customers A and B; only A has 2 orders. Keep both with LEFT JOIN.
- GROUP BY customer key; COUNT(OrderID) yields A:2, B:0.
- COUNT(*) would count B's unmatched joined row as 1.
- Avoid: COUNT(*) for zero child count; WHERE for aggregate totals

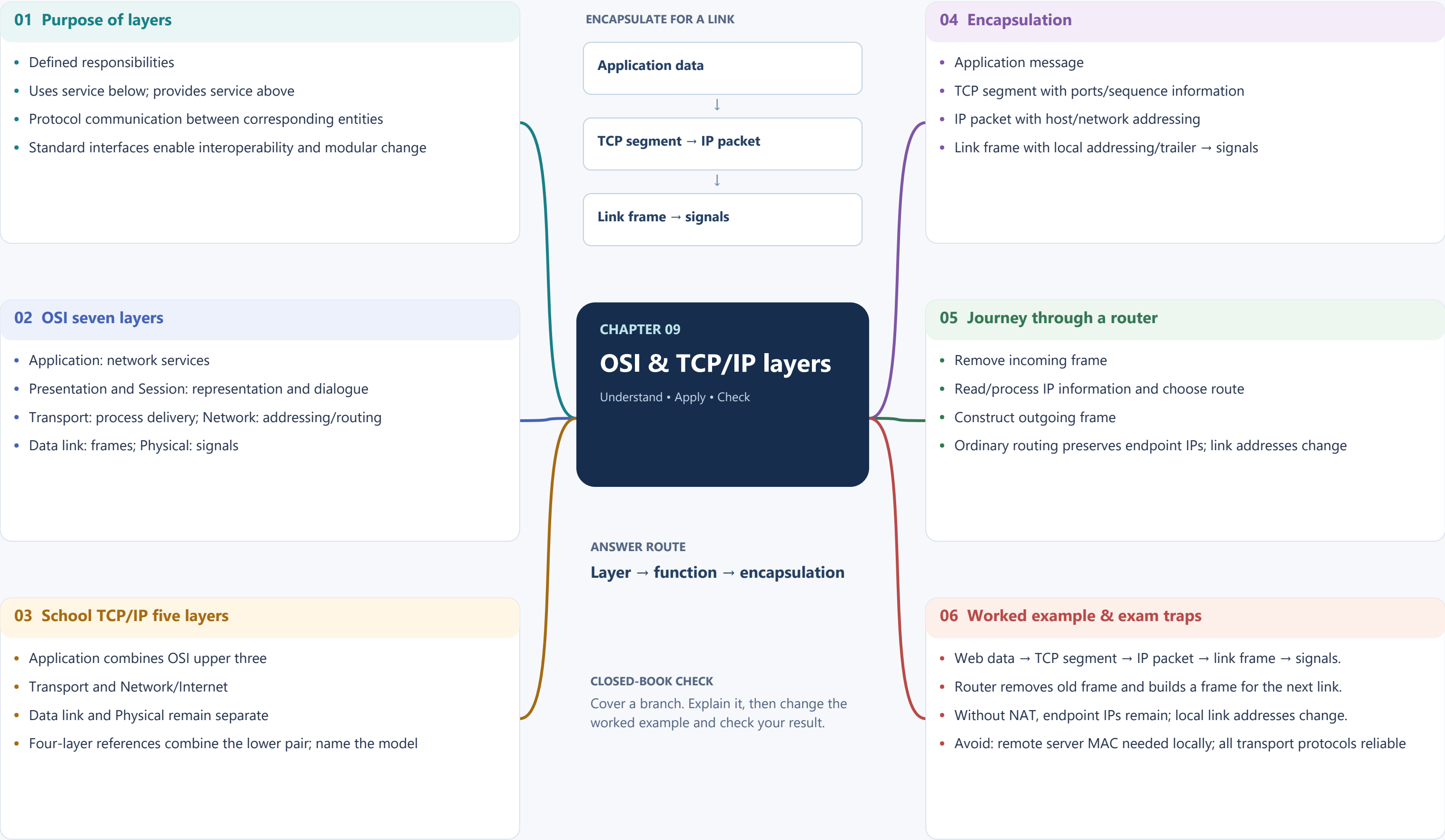
NoSQL theory



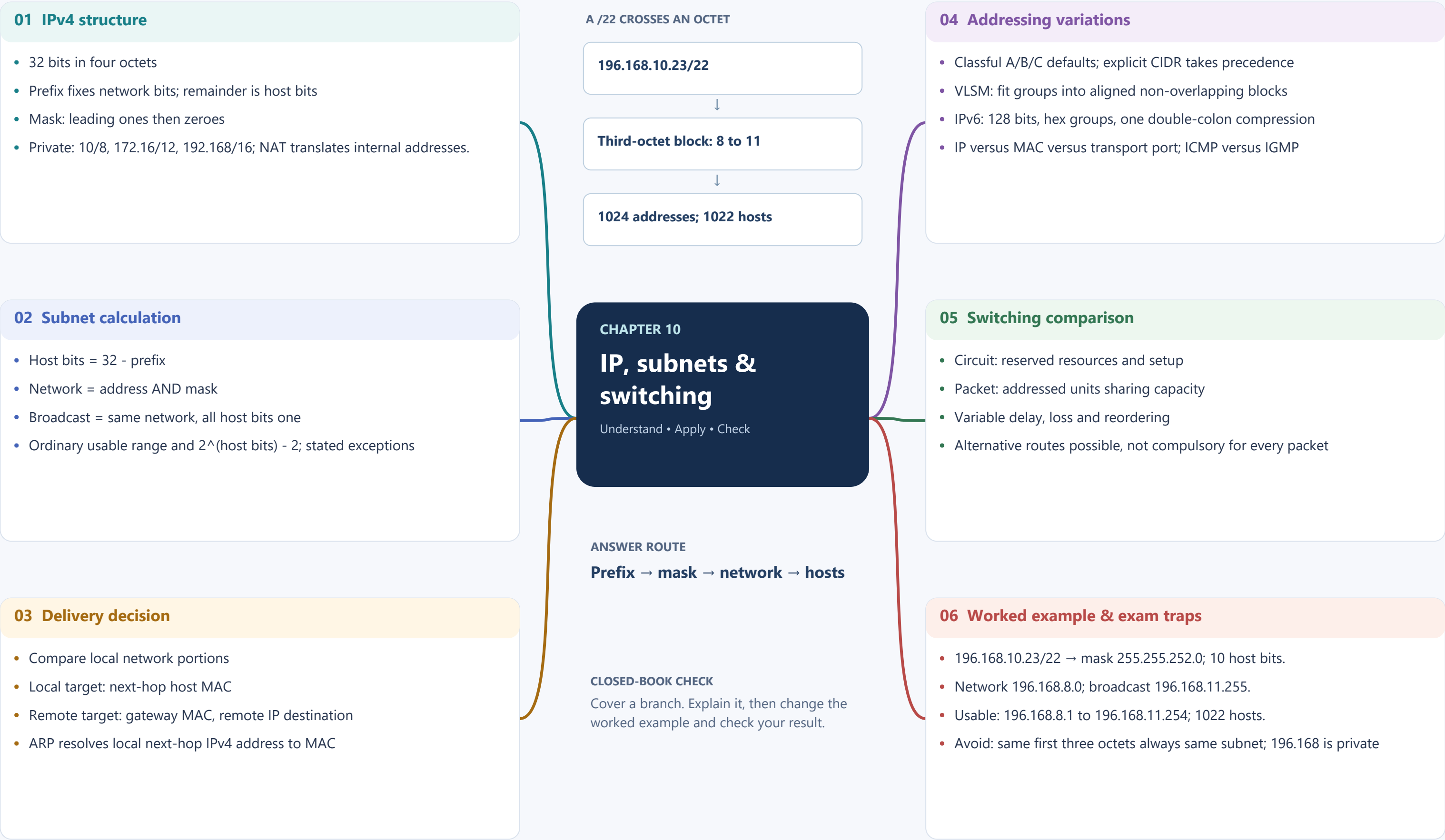
Network foundations, hardware and architectures



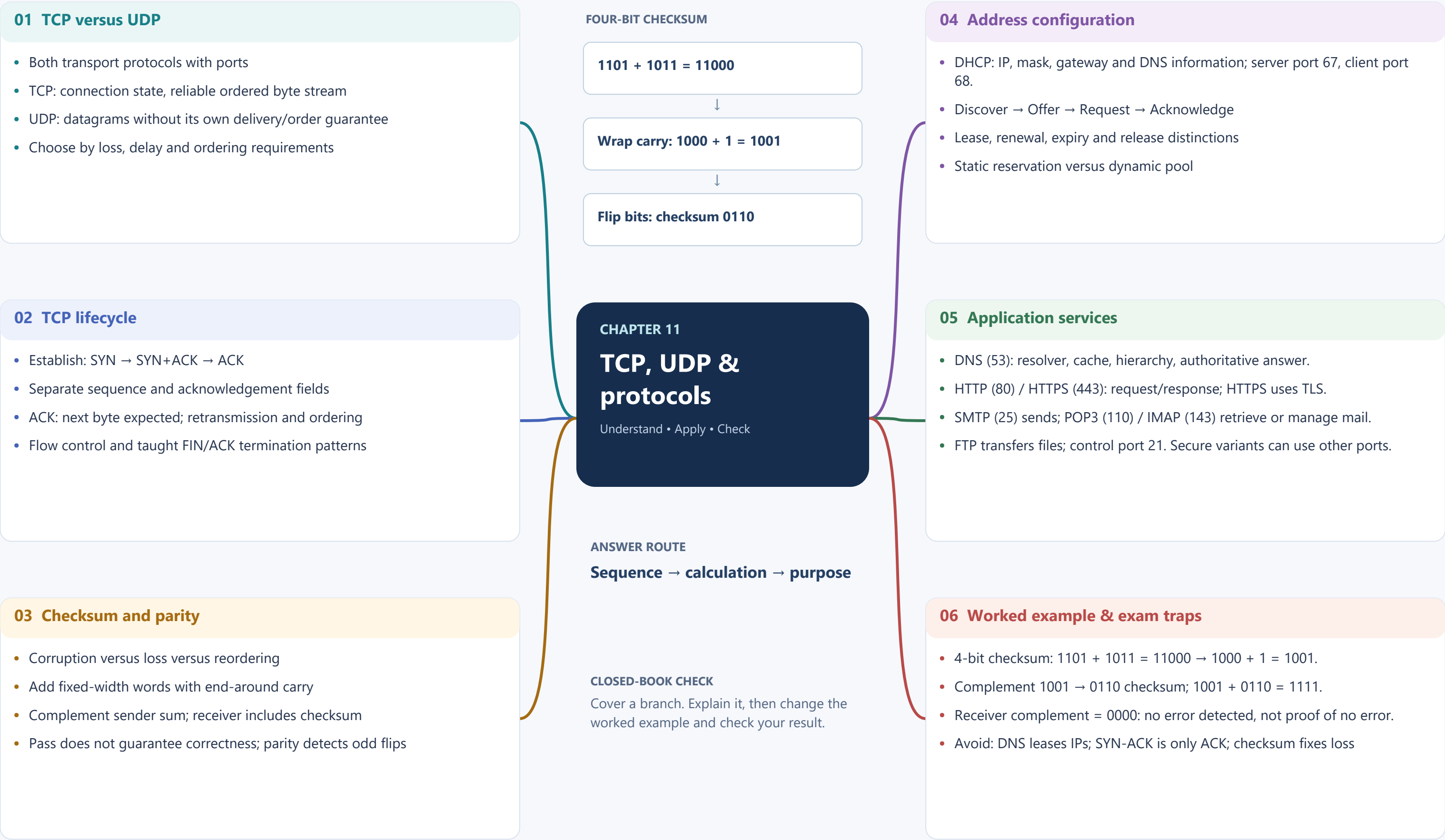
OSI and TCP/IP layers



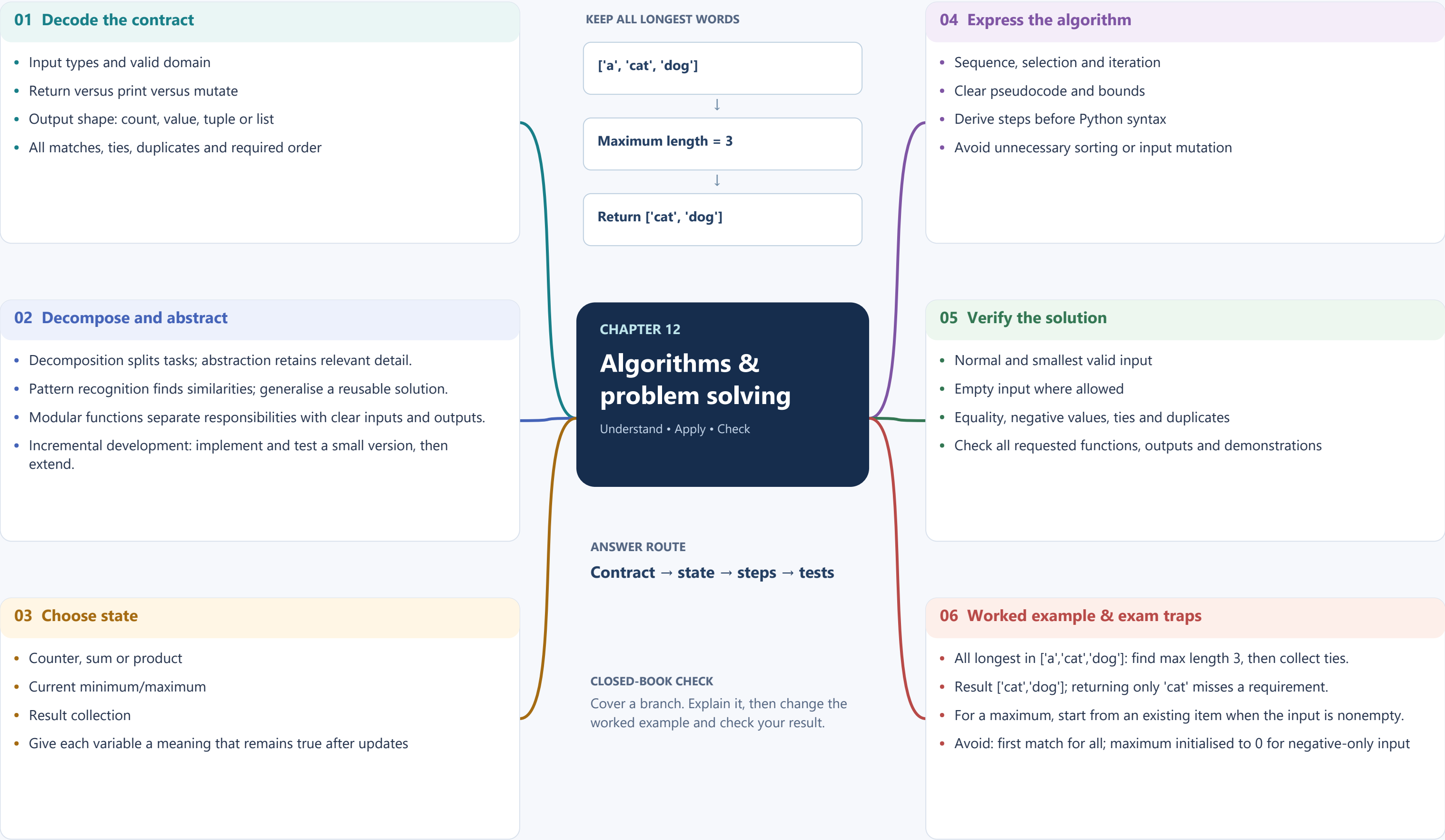
IP addressing, subnetting and switching



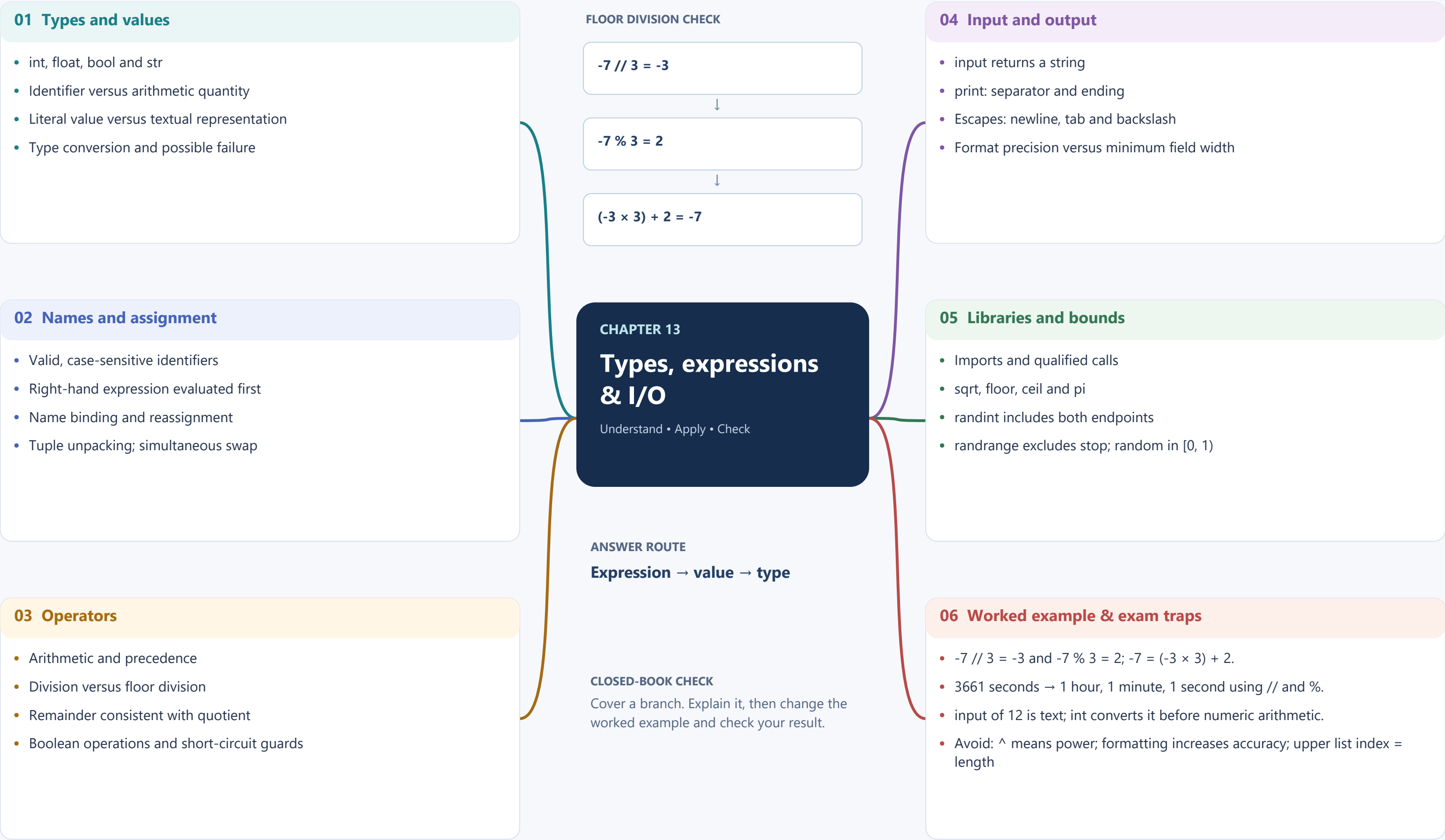
TCP, UDP, checksums and application protocols



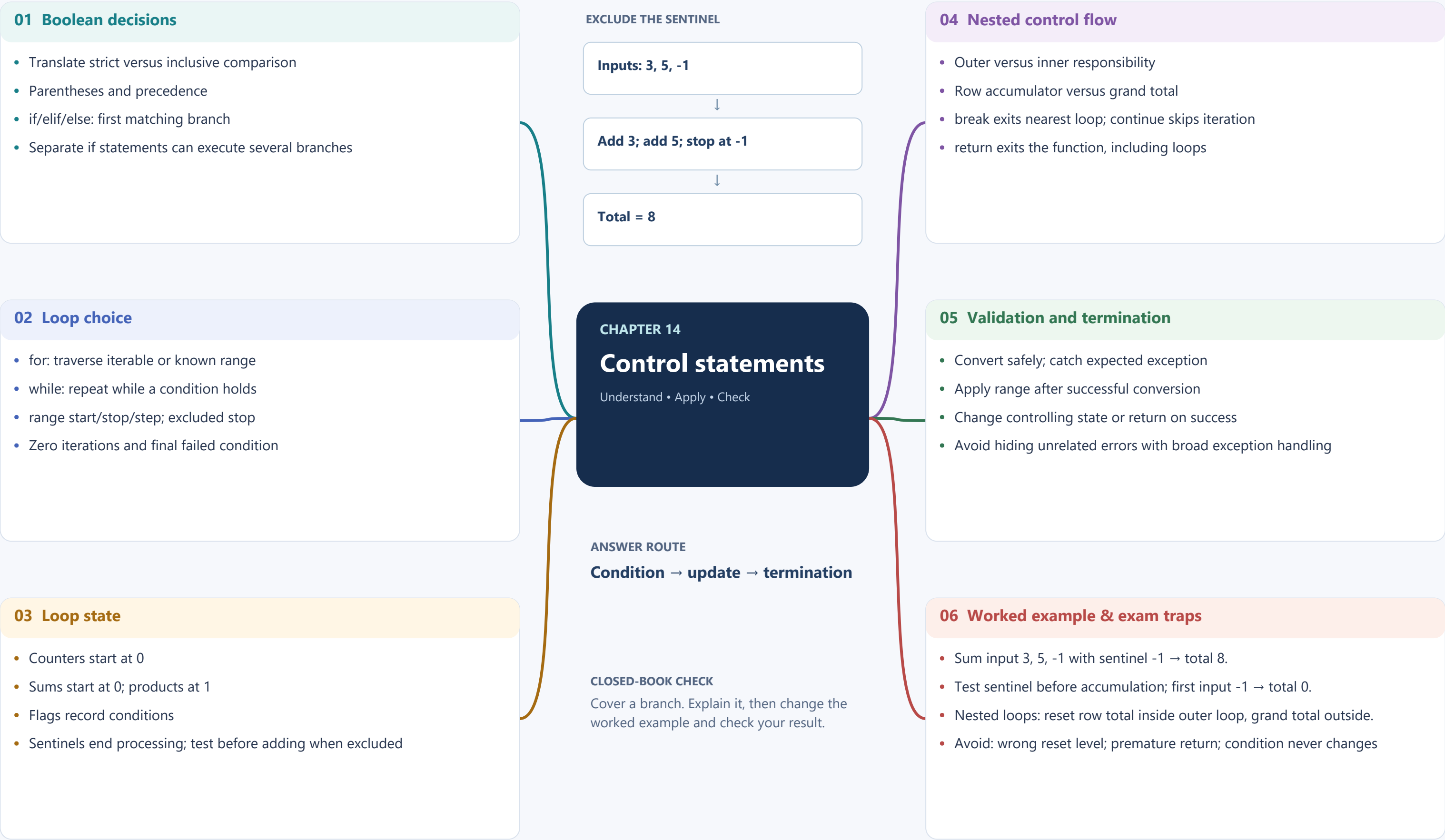
Python Chapter 1 - algorithms and problem solving



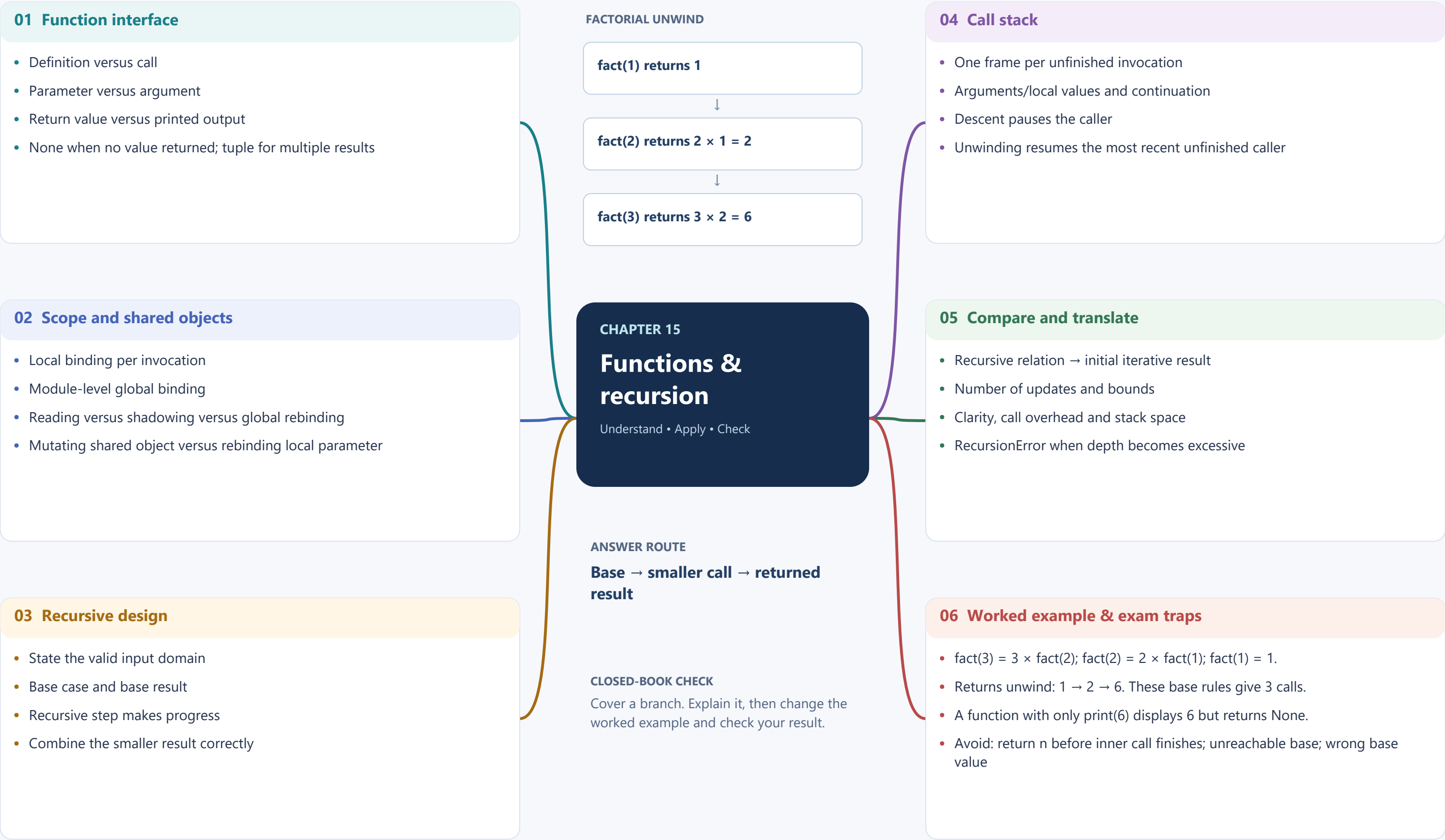
Python Chapter 2 - types, expressions and I/O



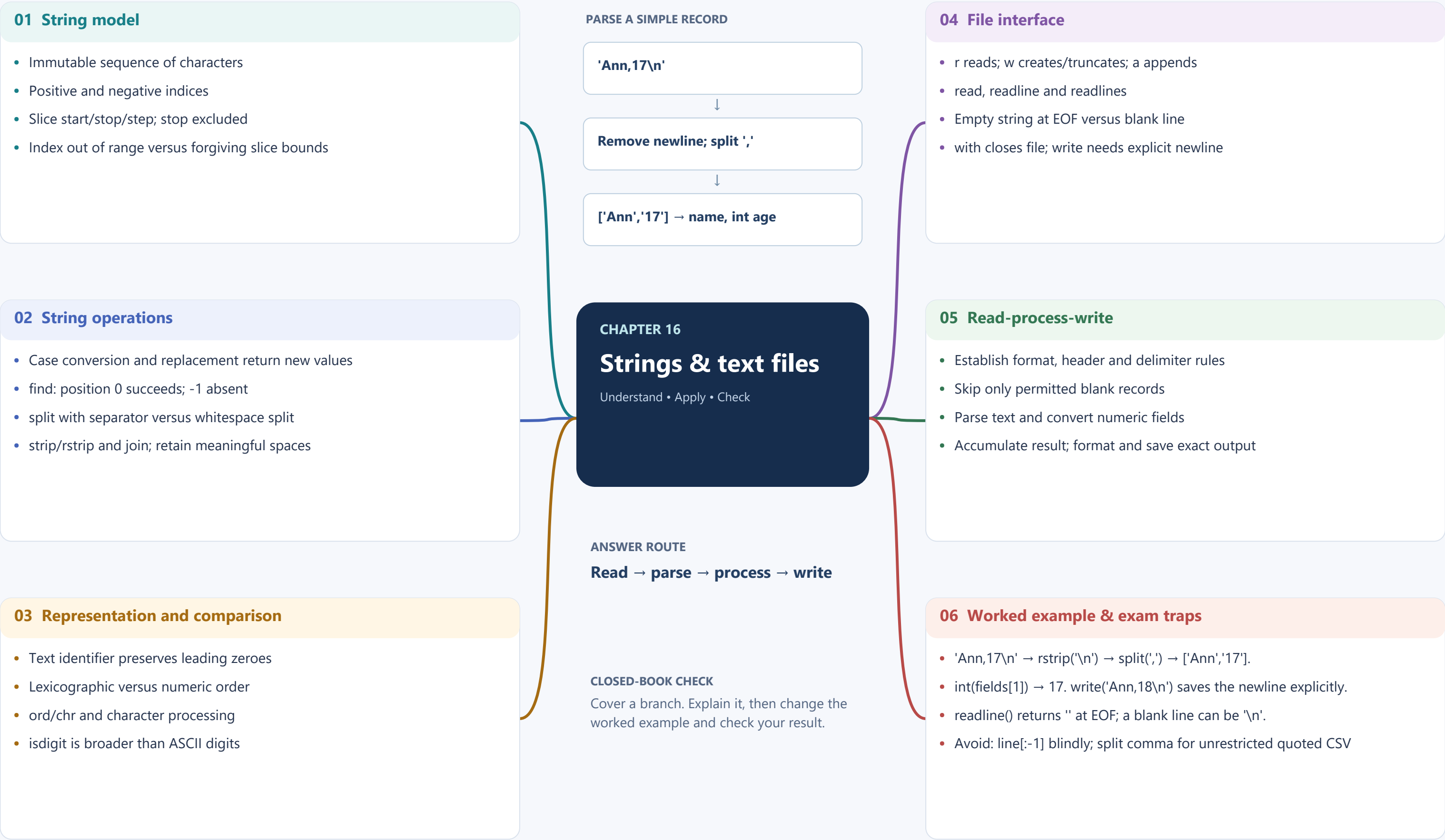
Python Chapter 3 - control statements



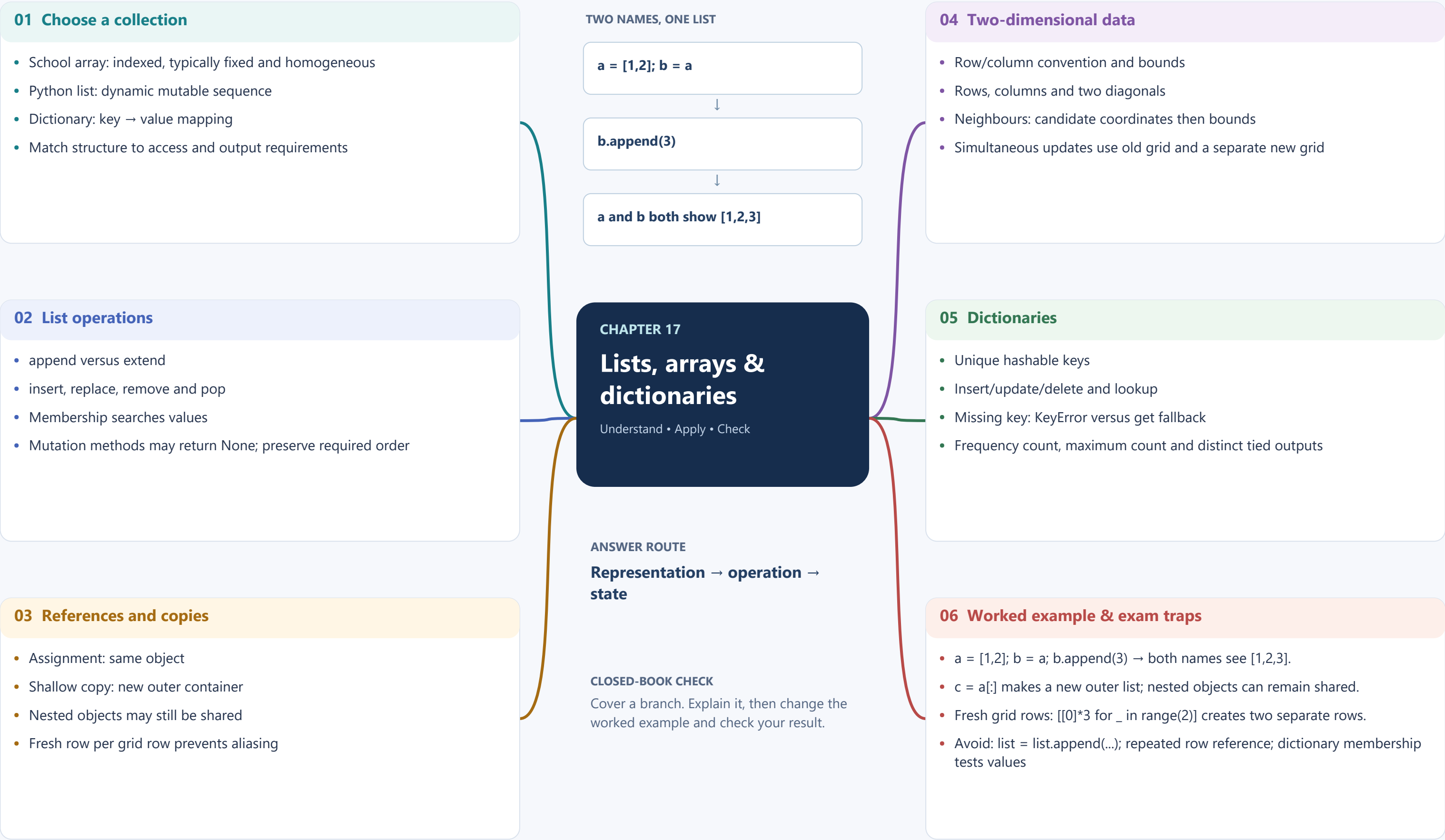
Python Chapter 4 - functions and recursion



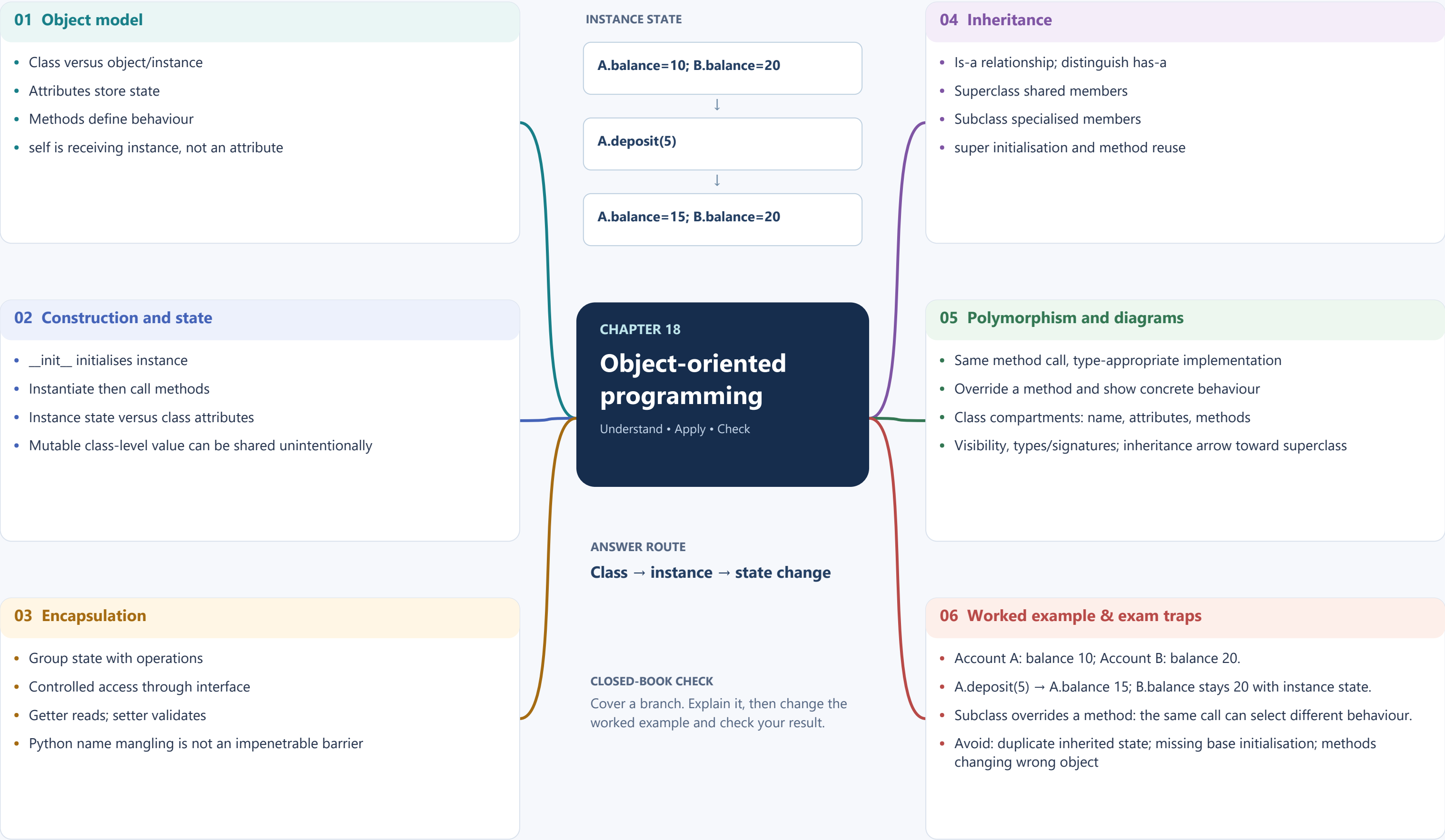
Python Chapter 5 - strings and text files



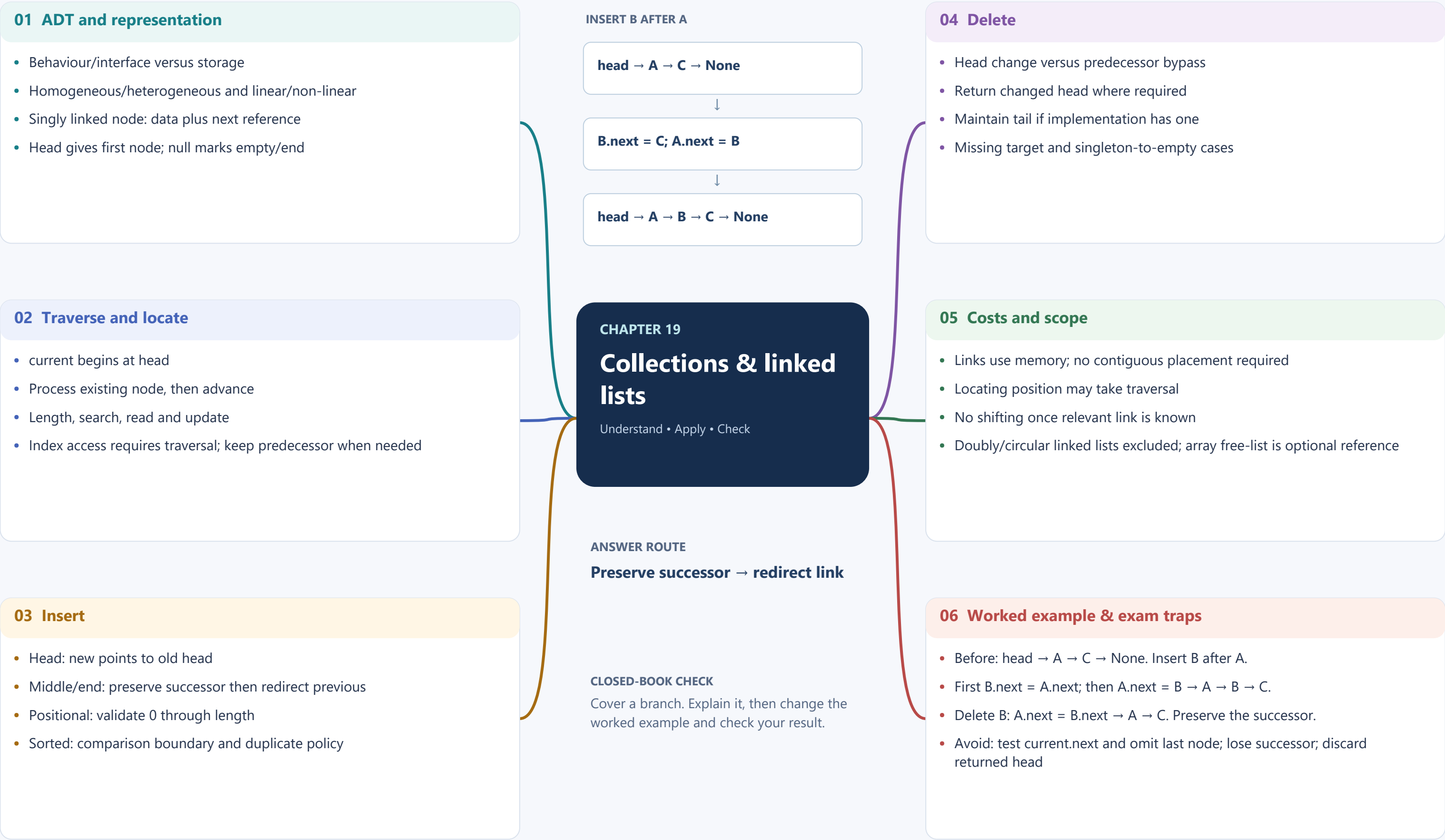
Python Chapter 6 - arrays, lists and dictionaries



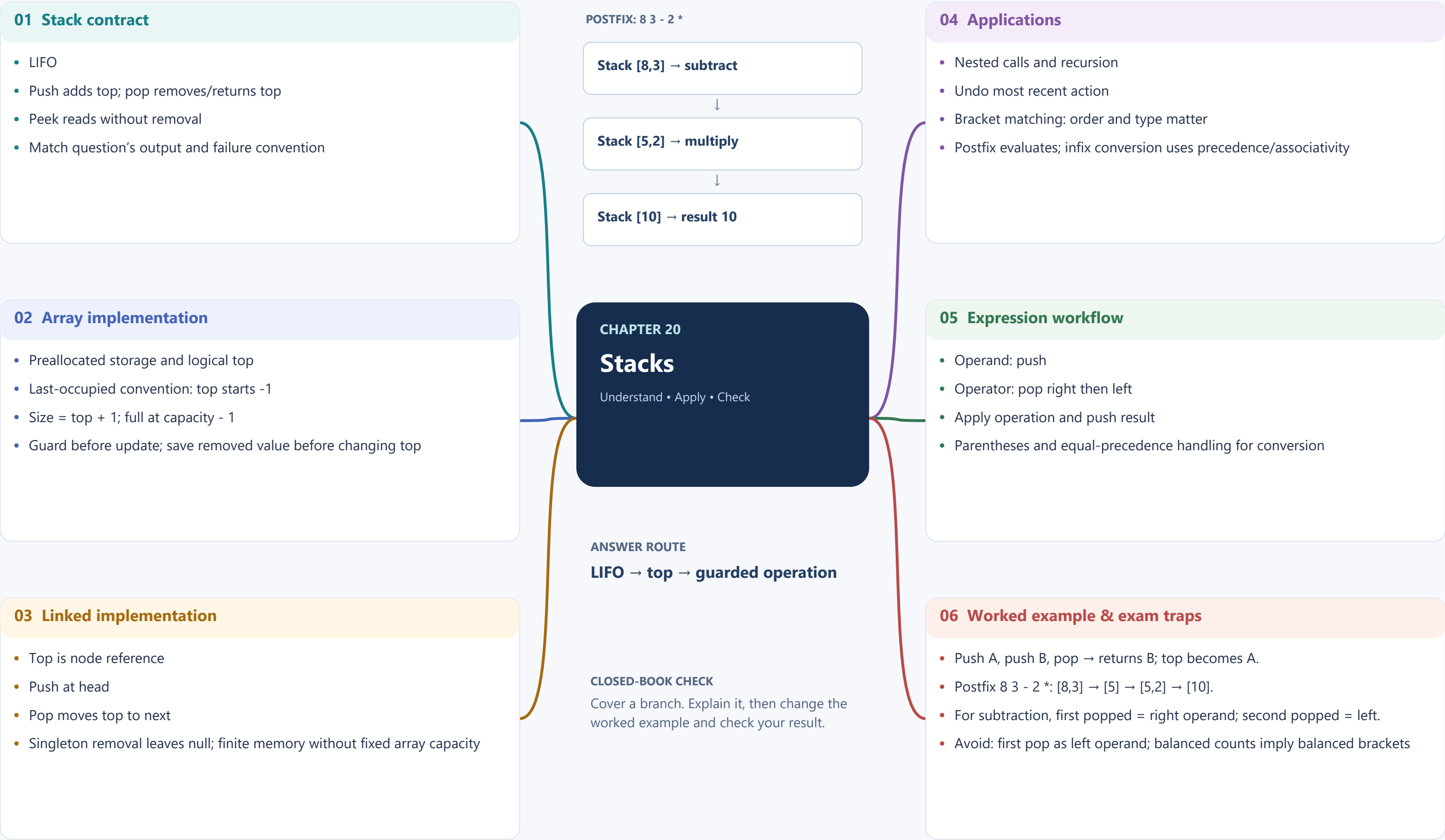
Python Chapter 7 - object-oriented programming



Python Chapter 8.1-8.3 - collections and linked lists



Python Chapter 8.4 - stacks



Python Chapter 8.5 - linear, linked and circular queues

