📅 Day 18: Learning Summary

Aptitude: Coding & Decoding

Coding-Decoding tests the ability to decipher the rule that transforms a word/number into a code.

Common Patterns

- **Letter Shifting:** Each letter is moved forward or backward by a fixed number (e.g., $A \to C$, $B \to D$ is a $+2$ shift).
- **Reverse Mapping:** Letters are mapped to their opposites in the alphabet ($A \leftrightarrow Z$, $B \leftrightarrow Y$).
  - *Quick Tip:* The sum of the positions of opposite letters is always 27 (e.g., $A(1) + Z(26) = 27$).
- **Numerical Coding:** Words are replaced by numbers based on their alphabetical position or a custom key.
- **Symbol Coding:** Assigning specific symbols to letters (e.g., $T = \#$, $A = @$).

Programming: Find Missing Number in an Array

Given an array containing $n - 1$ integers in the range of 1 to n, find the one that is missing.

Approach 1: Sum Formula (Most Efficient)

- **Logic:** Calculate the sum of the first n natural numbers and subtract the sum of the elements present in the array.
- **Formula:** $\text{Sum} = \frac{n(n+1)}{2}$
- **Complexity:** $O(n)$ time, $O(1)$ space.

Approach 2: XOR Method

- **Logic:** XOR all numbers from 1 to n, then XOR that result with all elements in the array.
- **Why it works:** $x \oplus x = 0$ and $x \oplus 0 = x$. The elements present in the array will cancel out, leaving only the missing number.
- **Benefit:** Prevents integer overflow which can happen with the Sum Formula in very large arrays.

Concept: Python Abstraction

Abstraction is the process of hiding internal complexity and showing only the necessary features of an object.

- **Abstract Base Classes (ABC):** Python uses the `abc` module to define abstract classes.

- **Implementation:**

```
from abc import ABC, abstractmethod

class Shape(ABC):
    @abstractmethod
    def area(self):
        pass

class Square(Shape):
    def __init__(self, side):
        self.side = side
    def area(self):
        return self.side * self.side
```

- **Key Point:** You cannot instantiate an abstract class (`Shape()`). It serves as a blueprint for other classes.

C++ Concept: Abstract Classes

In C++, an abstract class is defined by the presence of at least one **Pure Virtual Function**.

Pure Virtual Function

A function that has no implementation in the base class and is declared using `= 0` .

```
class Vehicle {
public:
    virtual void startEngine() = 0; // Pure virtual function
};
```

Interface vs. Abstract Class

- **Abstract Class:** Can have both normal functions (with code) and pure virtual functions.
- **Interface (Conceptual):** A class where *all* functions are pure virtual. It enforces a strict contract that derived classes must follow.

📊 SQL: HAVING Clause

The `HAVING` clause was added to SQL because the `WHERE` keyword could not be used with aggregate functions.

Key Differences: WHERE vs. HAVING

Feature	WHERE	HAVING
Applied To	Individual rows	Groups (after `GROUP BY`)

Usage	Filters data before grouping	Filters data after grouping
Aggregates	Cannot contain aggregates (e.g., `SUM`)	Can contain aggregate functions

Practical Example

If you want to find departments where the *total* salary expenditure is over $\$200,000$:

```
SELECT Department, SUM(Salary)
FROM Employees
GROUP BY Department
HAVING SUM(Salary) > 200000;
```

If you used `WHERE SUM(Salary) > 200000`, the database would throw an error because `WHERE` looks at rows one by one, and a single row doesn't have a "sum" yet.