Day 3: Aptitude, Programming & Core Concepts

1. Aptitude: Time & Work (Efficiency)

Time and Work problems often revolve around the relationship between the time taken to complete a task and the efficiency (rate of work) of the person or machine.

Key Concepts

- **Inverse Relationship:** Efficiency is inversely proportional to the time taken. If Efficiency increases, Time taken decreases.
  - $Efficiency \propto \frac{1}{Time}$
- **Work Formula:** $Work = Efficiency \times Time$
- **Ratio Method:** If the ratio of efficiency of A and B is $x : y$, then the ratio of time taken by them to finish the same work is $y : x$.
- **Combined Efficiency:** If A can do a work in D_1 days and B in D_2 days, their one-day work is $\frac{1}{D_1} + \frac{1}{D_2}$.

2. Programming: Reverse a Number

Reversing a number follows a similar logic to finding the "Sum of Digits" but involves building a new number by shifting decimal places.

Logic

1. Initialize `reversed_num = 0`.
2. Extract the last digit: `digit = num % 10`.
3. Update `reversed_num`: `reversed_num = (reversed_num * 10) + digit`.
4. Remove the last digit from the original: `num //= 10`.
5. Repeat until `num` is 0.

Python Implementation

```python
def reverse_number(n):
    rev = 0
    while n > 0:
        rev = (rev * 10) + (n % 10)
        n //= 10
    return rev

print(reverse_number(1234)) # Output: 4321
```

3. Python: Conditional Statements

Python uses indentation to define blocks of code that execute based on whether a condition is `True` or `False` .

- `if` **statement:** Executes a block if the condition is true.

- `elif` **(else if):** Checks another condition if the previous ones were false.

- `else` **:** Executes if none of the above conditions are met.

Syntax:

```
age = 20
if age >= 18:
    print("Adult")
elif age > 12:
    print("Teenager")
else:
    print("Child")
```

4. C/C++: Conditional Branching

C++ handles branching using `if-else` and `switch` statements. Unlike Python, C++ uses curly braces `{}` to define code blocks.

- `if...else if...else` : Standard logical branching.

- `switch` **statement**: Useful for comparing a single variable against multiple constant values (cases).

Example:

```
int choice = 2;
switch(choice) {
    case 1: cout << "One"; break;
    case 2: cout << "Two"; break; // This will execute
    default: cout << "Invalid";
}
```

5. SQL: Data Definition Language (DDL)

`CREATE` and `DROP` are part of DDL, used to define or modify the structure (schema) of the database.

CREATE TABLE

Used to create a new table and define its columns and data types.

```
CREATE TABLE Students (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
    EnrollmentDate DATE
```

```
    );
```

DROP TABLE

Used to permanently delete an existing table and all its data. **Warning:** This cannot be undone.

```
DROP TABLE Students;
```

6. Practice Questions

1. **Aptitude:** A is thrice as efficient as B. If B takes 24 days to finish a task, how many days will A take?

2. **Programming:** Modify the "Reverse a Number" logic to check if a number is a **Palindrome** (a number that reads the same backward as forward).

3. **Python:** What is the difference between `if-if-if` and `if-elif-else` in terms of execution?

4. **C/C++:** Why is the `break` statement important in a `switch` block? What happens if it is omitted?

5. **SQL:** Write a SQL command to create a table named `Employees` with columns for `EmpID` (Integer), `Salary` (Decimal), and `JobTitle` (Text).

6. **Aptitude:** If 10 men can complete a piece of work in 15 days, how many men are required to finish the same work in 10 days?

7. **Programming:** In the number reversal logic, what happens if the input number ends in a zero (e.g., `120`)?

8. **Logic:** Which logical operator (`&&` or `||`) is used in C++ to check if *at least one* of two conditions is true?

9. **SQL:** What is the difference between `DROP TABLE` and `DELETE FROM TABLE` ?

10. **Syntax:** In C++, can you use a `float` variable as the expression in a `switch` statement?