

## Day 5: #60-Day Placement Sprint Challenge

### 1. Aptitude: Simple Interest (SI)

Simple Interest is calculated only on the principal amount, or on that portion of the principal amount that remains.

#### Core Formulas

- **Simple Interest (SI):**  $SI = \frac{P \times R \times T}{100}$
- **Amount (A):**  $A = P + SI = P(1 + \frac{RT}{100})$
- Where:
  - **P** = Principal (Initial sum)
  - **R** = Rate of Interest (per annum)
  - **T** = Time (in years)

#### Key Shortcuts & Concepts

1. **Time Calculation:** If time is given in days, divide by 365. If in months, divide by 12.
2. **Sum becomes 'n' times:** If a sum of money becomes  $n$  times itself in  $T$  years, the rate of interest is  $R = \frac{100(n-1)}{T}$ .
3. **Changing Rates:** If the rate changes from  $R_1$  to  $R_2$  on the same principal, the change in interest is  $\Delta SI = \frac{P \times (R_1 - R_2) \times T}{100}$ .

### 2. Programming: Fibonacci Series (Using Loops)

The Fibonacci series is a sequence where each number is the sum of the two preceding ones, starting from 0 and 1. **Sequence:** 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

#### Logic

1. Initialize  $a = 0$ ,  $b = 1$ .
2. Print  $a$  and  $b$ .
3. Loop from 2 to  $n$ :
  - $next\_term = a + b$
  - Update  $a = b$
  - Update  $b = next\_term$

#### Implementation (Python)

```
def fibonacci_loop(n):
    a, b = 0, 1
```

```

if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print(a)
else:
    print(a, b, end=" ")
    for i in range(2, n):
        next_term = a + b
        print(next_term, end=" ")
        a = b
        b = next_term

# Example: Generate first 10 terms
fibonacci_loop(10)

```

### 3. Concept: Python Functions

Functions are "first-class citizens" in Python, allowing for modularity and reusability.

#### Key Aspects

- **Modular Coding:** Breaking a large codebase into smaller, independent chunks.
- **DRY Principle:** "Don't Repeat Yourself." Write once, call many times.
- **Arguments vs. Parameters:**
  - **Parameters:** Variables listed in the function definition (e.g., `def add(a, b):` ).
  - **Arguments:** Actual values passed when calling (e.g., `add(5, 3)` ).
- **Default Arguments:** Python allows functions to have default values (e.g., `def greet(name="Guest"):` ).

### 4. C/C++: Function Definition & Calling

In C/C++, functions require stricter structure compared to Python, often involving a declaration (prototype) and a definition.

#### The Three Components

1. **Function Declaration (Prototype):** Tells the compiler about the function name and parameters before it's used.
  - `int multiply(int a, int b);`
2. **Function Definition:** The actual body of the function.
  - `int multiply(int a, int b) { return a * b; }`
3. **Function Call:** Triggering the execution.
  - `int result = multiply(10, 20);`

#### Execution Flow

1. Control moves from `main()` to the function.

2. Local variables for the function are created in the **Stack**.
3. The function executes its body.
4. The `return` statement sends the result back to the caller and destroys local variables.

## 5. SQL: WHERE Clause

The `WHERE` clause is used to filter records. It is used to extract only those records that fulfill a specified condition.

### Syntax

```
SELECT column1, column2
FROM table_name
WHERE condition;
```

### Common Operators

- **Comparison:** `=`, `<>`, `!=`, `<`, `>`, `<=`, `>=`
- **Range:** `BETWEEN ... AND ...`
- **Pattern Matching:** `LIKE` (e.g., `WHERE name LIKE 'A%`')
- **List Search:** `IN` (e.g., `WHERE city IN ('Delhi', 'Mumbai')`)
- **Logical:** `AND`, `OR`, `NOT`

### Practice Questions

1. **Basic Calculation:** Find the Simple Interest on \$12,500 at 6% per annum for 3 years and 6 months.
2. **Rate Finding:** A sum of \$1,200 amounts to \$1,536 in 4 years at simple interest. What is the annual rate of interest?
3. **Shortcut Challenge:** A certain sum of money becomes triple (3 times) itself in 10 years. In how many years will it become 5 times itself at the same rate of interest?
4. **Scope Identification:** If a variable `x = 10` is defined outside a function, and we define `x = 20` inside the function, what will be the value of `x` printed outside the function after the function call?
5. **Default Arguments:** Given `def power(a, b=2): return a**b`, what are the outputs for `power(3)` and `power(3, 3)`?
6. **Prototyping:** Why do we get a "function not declared" error in C++ if we call a function in `main()` but define it below the `main()` block without a prototype?
7. **Memory:** Explain why local variables of a function are "lost" once the function finishes execution.

11. **Filtering:** Write a query to fetch `EmpName` and `Salary` from the `Employees` table for those who earn more than 40,000 but less than 70,000 (using `BETWEEN` ).
12. **Pattern Matching:** Write an SQL query to find all students in a `Students` table whose `Name` starts with 'S' and ends with 'n' (e.g., "Steven", "Sultan").