

1. Aptitude: Percentages

Percentages are the backbone of many aptitude topics like Profit & Loss, SI/CI, and Data Interpretation.

- **Core Formula:** $\text{Percentage} = \left(\frac{\text{Value}}{\text{Total Value}} \right) \times 100$
- **Percentage Change:** $\frac{\text{Final} - \text{Initial}}{\text{Initial}} \times 100$
- **Successive Percentage Change:** If a value is increased by $a\%$ and then by $b\%$, the net change is:

$$\left(a + b + \frac{ab}{100} \right) \%$$

Quick Conversion Table

Fraction	Percentage	Fraction	Percentage
1/2	50%	1/6	16.66%
1/3	33.33%	1/7	14.28%
1/4	25%	1/8	12.5%
1/5	20%	1/9	11.11%

2. Programming: Python Pattern Printing

Pattern printing is essential for mastering **Nested Loops**. The outer loop typically controls the rows, while the inner loop handles the columns (or the content within the row).

Basic Square Pattern

```
n = 5
for i in range(n):
    for j in range(n):
        print("*", end=" ")
    print() # Moves to next line after row completion
```

Right-Angled Triangle

```
n = 5
for i in range(1, n + 1):
    for j in range(i):
        print("*", end=" ")
    print()
```

```
print()
```

Inverted Triangle (Logic Check)

To print an inverted triangle, the inner loop range must decrease as the outer loop index increases.

```
n = 5
for i in range(n, 0, -1):
    print("* " * i)
```

3. Concept: Python Basics

Variables & Data Types

Python is **dynamically typed**, meaning the interpreter determines the type at runtime based on the value assigned.

- **int:** Whole numbers (e.g., `count = 5`)
- **float:** Decimal numbers (e.g., `pi = 3.14`)
- **str:** Text wrapped in quotes (e.g., `course = "Placement Sprint"`)
- **bool:** Binary logic (e.g., `is_passed = True`)

Control Flow: Loops

- **For Loop:** Best for fixed iterations or iterating over sequences.
- **While Loop:** Best when the number of iterations is unknown and depends on a condition.
- **Break/Continue:** `break` exits the loop entirely; `continue` skips the current iteration.

4. C/C++ Concept: Data Types & I/O

C/C++ is **statically typed** and requires memory management awareness.

Primitive Data Types

Type	Size (Typical)	Range
int	4 bytes	-2^{31} to $2^{31} - 1$
float	4 bytes	~7 decimal digits
char	1 byte	-128 to 127 (ASCII)
double	8 bytes	~15 decimal digits

Input/Output (Syntax Comparison)

- **C:** Uses `stdio.h`. `scanf` requires address-of operator (`&`).

```
int age;
scanf("%d", &age);
printf("Age is %d", age);
```

- **C++:** Uses `iostream`. `cin` and `cout` use stream operators (`>>` and `<<`).

```
int age;
std::cin >> age;
std::cout << "Age is " << age;
```

5. SQL: Introduction to Databases

Core Terminology

- **RDBMS:** Relational Database Management System (e.g., MySQL, PostgreSQL).
- **Schema:** The blueprint or structure of the database.
- **Primary Key:** A unique identifier for each record in a table (cannot be NULL).

Basic SQL Commands (CRUD Operations)

1. **CREATE:** Define a new table.

```
CREATE TABLE Students (
    ID int PRIMARY KEY,
    Name varchar(50),
    Major varchar(50)
);
```

2. **INSERT:** Add data.

```
INSERT INTO Students VALUES (1, 'Rahul', 'Computer Science');
```

3. **SELECT:** Retrieve data.

```
SELECT Name FROM Students WHERE Major = 'Computer Science';
```

4. **UPDATE:** Modify existing data.

```
UPDATE Students SET Major = 'IT' WHERE ID = 1;
```

6. Practice Challenges (Day 1)

1. **Aptitude:** A student's score increased from 60 to 75. Calculate the percentage increase.
2. **Programming:** Write a Python program to print a 5×5 square of numbers where each row contains the row number (e.g., Row 1 is all 1s).
3. **SQL:** Write a query to find all students from a table named `Employees` where the `Salary` is greater than 50,000.