Day 16 Study Notes

1. Aptitude: Time & Clocks

Clocks are a classic aptitude topic focusing on the relationship between time, angles, and speed.

Key Concepts

- **Minute Hand:** Moves 6° per minute.
- **Hour Hand:** Moves 0.5° per minute (30° per hour).
- **Relative Speed:** The minute hand gains 5.5° ($6^\circ - 0.5^\circ$) over the hour hand every minute.

Angle Formula

To find the angle θ between the hour and minute hands at H hours and M minutes:

$$\theta = |30H - 5.5M|$$

Note: If the result is greater than 180°, subtract it from 360° to find the reflex angle.

2. Programming: Binary Search

Binary Search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing the search interval in half.

Complexity

- **Time Complexity:** $O(\log n)$
- **Space Complexity:** $O(1)$ (Iterative) or $O(\log n)$ (Recursive due to stack).

Pseudocode (Iterative)

```
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1

    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return f"target found at index{mid}"
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return "not found"
arr=[2,89,4,6]
arr.sort()
target=2
```

```
print(binary_search(arr,target))
```

3. Python & C++ Concept: Inheritance

Inheritance allows a class (Derived/Child) to inherit attributes and methods from another class (Base/Parent).

Key Benefits

1. **Reusability:** Use existing code without rewriting.
2. **Transitivity:** If B inherits from A, and C inherits from B, C also inherits from A.

Syntax Comparison

Python:

```
class Animal:
    def speak(self):
        print("Animal makes a sound")

class Dog(Animal):
    def bark(self):
        print("Dog barks")
#Object
d = Dog()
d.speak()
d.bark()
```

C++:

```
class Parent {
public:
    void speak() { cout << "Parent speaking"; }
};

class Child : public Parent {
public:
    void talk() { cout << "Child talking"; }
};
```

4. SQL: Aggregate Functions

Aggregate functions perform a calculation on a set of values and return a single value. They are often used with the `GROUP BY` clause.

Function	Description

`COUNT()`	Returns the number of rows.
`SUM()`	Returns the total sum of a numeric column.
`AVG()`	Returns the average value of a numeric column.
`MIN()`	Returns the smallest value.
`MAX()`	Returns the largest value.

Example Query

```
SELECT department_id, COUNT(employee_id), AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) > 50000;
```

Note: Use `HAVING` instead of `WHERE` to filter results based on aggregate functions.

;