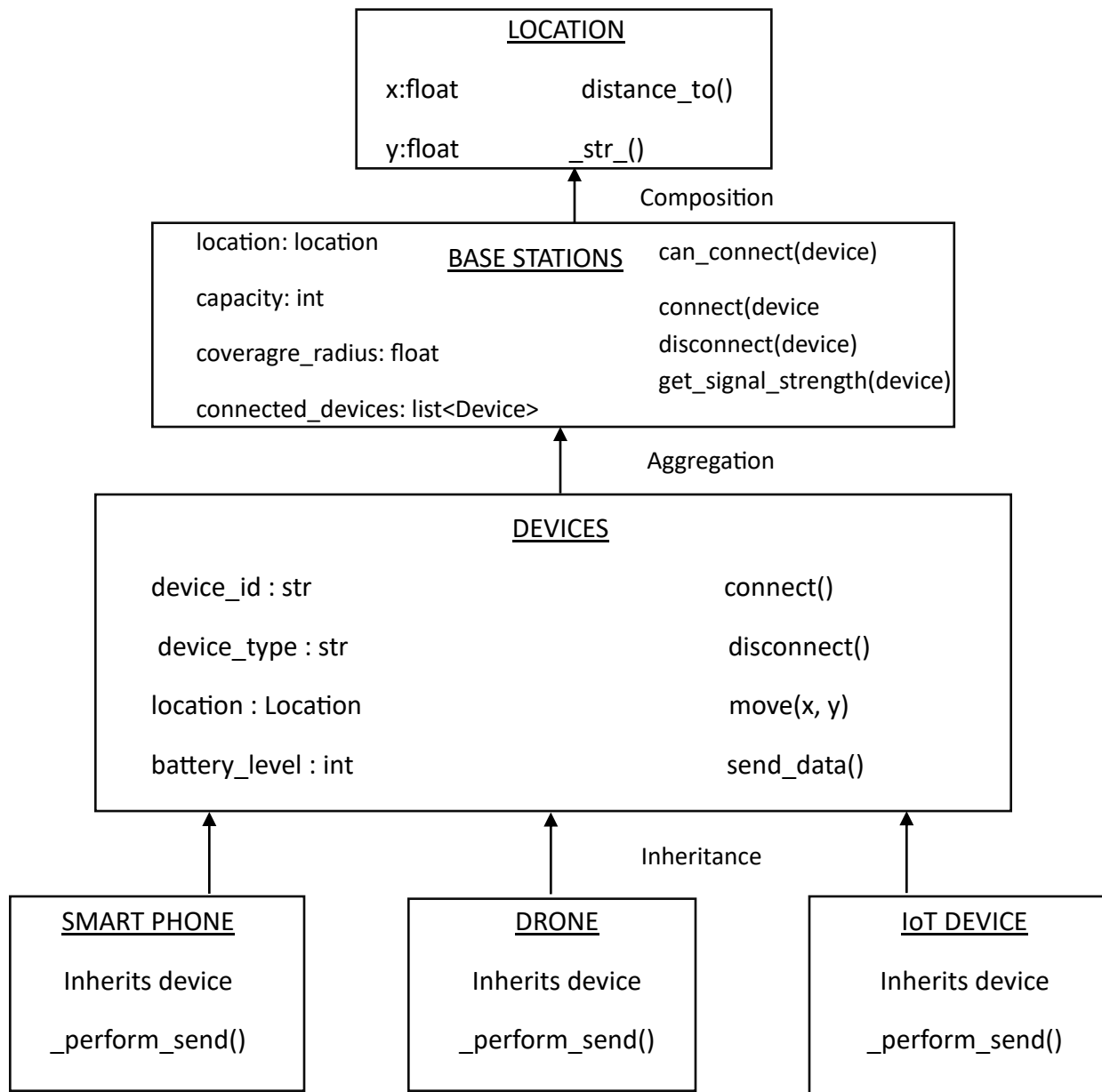


## TIE OOP- GROUP 4 : CLASS FLOW DIAGRAM



❖ Base station contains location(composition) and manages multiple devices (aggregation)

## GUI INSTRUCTION MANUAL

### 1. System Requirements

Install Python 3.8 or higher

Works on Windows, Linux, or macOS

No internet connection required

## **2. Setup Instructions**

Open Notepad and paste the simulator Python code

Save the file as: mini\_5g\_simulation.py

Set Save as type → All Files

## **3. Running the Simulation**

Open Command Prompt ,navigate to the file location: cd Desktop\mini\_5g\_simulation

Run the program: python mini\_5g\_simulation.py

## **4. Simulation Walkthrough**

The simulation code: creates base stations with fixed coverage and capacity, registers different device types (SmartPhone, IoTDevice, Drone), automatically connects devices to available base stations, calculates signal strength based on distance and load, simulates data transmission using polymorphism, demonstrates mobility and automatic handover and shows encapsulated device information

## **5. Using the Simulatoion**

The simulation runs automatically. nThe user only observes output at the console and modifies device locations or add devices/base stations to test different scenarios

## **6. Key Features Demonstrated**

Object-Oriented Programming principles

Device mobility and handover

Signal strength management

Error handling for out-of-range and capacity limits

Encapsulation and polymorphism

## **7. Ending the Program**

The program ends automatically after completing the demo

## **ASSUMPTIONS AND LIMITATIONS OF THE SIMULATION**

## Assumptions

- Flat 2D environment
- Circular base station coverage
- Instant handover
- No packet loss modeling
- Constant battery level

## Limitations

- No real 5G protocol stack
- No interference modeling
- Console-only output
- No real-time simulation

## DEMO OUTPUT

```
Administrator: C:\Windows\S... x + v
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\mini_5g_simulation>python mini_5g_simulation.py
=== Mini 5G Network Simulator Demo ===

Base stations created:
BaseStation at (0, 0), connected: 0/5
BaseStation at (200, 0), connected: 0/3

Devices registered:
SmartPhone [PHONE-001] at (20, 20)
IoTDevice [IOT-001] at (30, 10)
Drone [DRONE-001] at (50, 50)

SmartPhone [PHONE-001] at (20, 20) attempting to connect...
SmartPhone [PHONE-001] at (20, 20) Connected successfully to BaseStation at (0, 0), connected: 1/5
IoTDevice [IOT-001] at (30, 10) attempting to connect...
IoTDevice [IOT-001] at (30, 10) Connected successfully to BaseStation at (0, 0), connected: 2/5
Drone [DRONE-001] at (50, 50) attempting to connect...
Drone [DRONE-001] at (50, 50) Connected successfully to BaseStation at (0, 0), connected: 3/5

SmartPhone [PHONE-001] at (20, 20) sending data via BaseStation at (0, 0), connected: 3/5
  SmartPhone: High-speed data transfer / call simulation
IoTDevice [IOT-001] at (30, 10) sending data via BaseStation at (0, 0), connected: 3/5
  IoTDevice: Low-power sensor data upload
Drone [DRONE-001] at (50, 50) sending data via BaseStation at (0, 0), connected: 3/5
  Drone: Streaming video data

Drone [DRONE-001] at (180, 10) moved from (50, 50) to (180, 10)
Signal lost due to move, triggering handover/disconnect
Drone [DRONE-001] at (180, 10) disconnected from None
Drone [DRONE-001] at (180, 10) attempting to connect...
Drone [DRONE-001] at (180, 10) Connected successfully to BaseStation at (200, 0), connected: 1/3
Drone [DRONE-001] at (180, 10) signal: 64.70%

Encapsulation demo:
{'battery_level': 100, 'device_id': 'PHONE-001', 'signal_strength': 57.37258300203048}

=== Demo complete ===

C:\Users\Administrator\Desktop\mini_5g_simulation>
```