Oops with C++

PROPERTY MANAGEMENT SYSTEM

Ojasw Lowanshi
Pratham Kashyap

PROGRAM :

```
include <iostream>

#include <vector>

#include <string>

using namespace std;


// Tenant Class

class Tenant {

public:

    string name;

    string phone;

    string email;


    Tenant(string name, string phone, string email)

        : name(name), phone(phone), email(email) {}


    void displayInfo() const {

        cout << "Tenant Name: " << name << "\nPhone: " << phone << "\nEmail: " << email <<
endl;

    }

};


// Base Property Class

class Property {

protected:

    int id;

    string address;
```

```
    double rent;

public:
    Property(int id, string address, double rent)
        : id(id), address(address), rent(rent) {}

    virtual void displayDetails() const = 0;
    int getId() const { return id; }
    double getRent() const { return rent; }
    string getAddress() const { return address; }

    virtual ~Property() {}
};

// Derived ResidentialProperty Class
class ResidentialProperty : public Property {
    int bedrooms;
public:
    ResidentialProperty(int id, string address, double rent, int bedrooms)
        : Property(id, address, rent), bedrooms(bedrooms) {}

    void displayDetails() const override {
        cout << "Residential Property ID: " << id
             << "\nAddress: " << address
             << "\nRent: $" << rent
             << "\nBedrooms: " << bedrooms << endl;
    }
```

```
};

// Derived CommercialProperty Class
class CommercialProperty : public Property {
    string businessType;
public:
    CommercialProperty(int id, string address, double rent, string businessType)
        : Property(id, address, rent), businessType(businessType) {}

    void displayDetails() const override {
        cout << "Commercial Property ID: " << id
             << "\nAddress: " << address
             << "\nRent: $" << rent
             << "\nBusiness Type: " << businessType << endl;
    }
};

// LeaseAgreement Class
class LeaseAgreement {
    Property* property;
    Tenant tenant;
    int durationMonths;

public:
    LeaseAgreement(Property* property, Tenant tenant, int duration)
        : property(property), tenant(tenant), durationMonths(duration) {}
```

```
    void displayLease() const {
        cout << "\n--- Lease Agreement ---" << endl;
        tenant.displayInfo();
        property->displayDetails();
        cout << "Duration: " << durationMonths << " months\nTotal Rent: $"
             << durationMonths * property->getRent() << endl;
    }
};

// Property Management System
class PropertyManagementSystem {
    vector<Property*> properties;
    vector<Tenant> tenants;
    vector<LeaseAgreement> leases;
    int propertyCounter = 1;

public:
    void addResidentialProperty(string address, double rent, int bedrooms) {
        properties.push_back(new ResidentialProperty(propertyCounter++, address, rent,
bedrooms));
    }

    void addCommercialProperty(string address, double rent, string businessType) {
        properties.push_back(new CommercialProperty(propertyCounter++, address, rent,
businessType));
    }

    void addTenant(string name, string phone, string email) {
```

```
        tenants.emplace_back(name, phone, email);
}

void createLease(int propertyId, int tenantIndex, int durationMonths) {
        Property* property = nullptr;
        for (auto* p : properties) {
                if (p->getId() == propertyId) {
                        property = p;
                        break;
                }
        }
        if (property && tenantIndex >= 0 && tenantIndex < tenants.size()) {
                leases.emplace_back(property, tenants[tenantIndex], durationMonths);
        } else {
                cout << "Invalid property ID or tenant index." << endl;
        }
}

void displayAllProperties() const {
        for (auto* p : properties) {
                p->displayDetails();
                cout << "----------" << endl;
        }
}

void displayAllTenants() const {
        for (const auto& t : tenants) {
```

```
                t.displayInfo();
                cout << "----------" << endl;
            }
        }

        void displayAllLeases() const {
            for (const auto& l : leases) {
                l.displayLease();
                cout << "----------" << endl;
            }
        }

        ~PropertyManagementSystem() {
            for (auto* p : properties) delete p;
        }
};

// Main Function
int main() {
    PropertyManagementSystem pms;

    // Adding properties
    pms.addResidentialProperty("123 Elm Street", 1200, 3);
    pms.addCommercialProperty("456 Market Ave", 2500, "Retail");

    // Adding tenants
    pms.addTenant("Alice Johnson", "555-1234", "alice@example.com");
```

```
    pms.addTenant("Bob Smith", "555-5678", "bob@example.com");

    // Creating leases
    pms.createLease(1, 0, 12);
    pms.createLease(2, 1, 6);

    // Display all
    pms.displayAllProperties();
    pms.displayAllTenants();
    pms.displayAllLeases();

    return 0;
}
```

OUTPUT

```
Residential Property ID: 1
Address: 123 Elm Street
Rent: $1200
Bedrooms: 3
---------------------
Commercial Property ID: 2
Address: 456 Market Ave
Rent: $2500
Business Type: Retail
---------------------
Tenant Name: Alice Johnson
Phone: 555-1234
Email: alice@example.com
---------------------
Tenant Name: Bob Smith
Phone: 555-5678
Email: bob@example.com
---------------------
```

```
--- Lease Agreement ---
Tenant Name: Alice Johnson
Phone: 555-1234
Email: alice@example.com
Residential Property ID: 1
Address: 123 Elm Street
Rent: $1200
Bedrooms: 3
Duration: 12 months
Total Rent: $14400
---------------------

--- Lease Agreement ---
Tenant Name: Bob Smith
Phone: 555-5678
Email: bob@example.com
Commercial Property ID: 2
Address: 456 Market Ave
Rent: $2500
Business Type: Retail
Duration: 6 months
Total Rent: $15000
```