**Question-1:**

#include <iostream>

#include <string>

class Employee {

protected:

int empID;

std::string empName;

double baseSalary;

public:

Employee(int id, const std::string& name, double salary)

: empID(id), empName(name), baseSalary(salary) {}

void displayEmployeeDetails() const {

std::cout << "Employee ID: " << empID << "\n";

std::cout << "Name: " << empName << "\n";

std::cout << "Base Salary: $" << baseSalary << "\n";

}

};

class Manager : virtual public Employee {

protected:

double bonus;

public:

Manager(int id, const std::string& name, double salary, double mgrBonus)

: Employee(id, name, salary), bonus(mgrBonus) {}

void displayManagerDetails() const {

displayEmployeeDetails();

std::cout << "Bonus: $" << bonus << "\n";

}

};

class TeamLeader : virtual public Employee {

protected:

int numTeamMembers;

public:

TeamLeader(int id, const std::string& name, double salary, int teamMembers)

: Employee(id, name, salary), numTeamMembers(teamMembers) {}

void displayTeamLeaderDetails() const {

displayEmployeeDetails();

std::cout << "Number of Team Members: " << numTeamMembers << "\n";

}

};

class SeniorManager : public Manager, public TeamLeader {

public:

SeniorManager(int id, const std::string& name, double salary, double mgrBonus, int teamMembers)

: Employee(id, name, salary), Manager(id, name, salary, mgrBonus), TeamLeader(id, name, salary, teamMembers) {}

void modifyAttributes(double newBonus, int newTeamMembers) {

bonus = newBonus;

numTeamMembers = newTeamMembers;

}

void displaySeniorManagerDetails() const {

displayEmployeeDetails();

std::cout << "Bonus: $" << bonus << "\n";

std::cout << "Number of Team Members: " << numTeamMembers << "\n";

}

};

int main() {

SeniorManager seniorMgr(101, "John Doe", 80000.0, 5000.0, 10);

seniorMgr.displaySeniorManagerDetails();

seniorMgr.modifyAttributes(6000.0, 12);

std::cout << "\nAfter Modification:\n";

seniorMgr.displaySeniorManagerDetails();

return 0;

}

**Question-2:**

#include <iostream>

#include <string>

using namespace std;

class Vehicle{

    protected:

    int vehicle\_ID;

    string brand;

    float rent\_rate;

    public:

    Vehicle(){

        cout << "Specify the Vehicle ID" << endl;

        cin >> vehicle\_ID;

        cout << "Specify Brand" << endl;

        cin >> brand;

        cout << "Rental per day: ";

        cin >> rent\_rate;

    }

    virtual void vehicleData() const{

        cout<< endl << "------Vehicle's Information-------" << endl;

        cout << "Vehicle ID: " << vehicle\_ID << endl;

        cout << "Brand: " << brand << endl;

        cout << "Rental per day: $" << rent\_rate << endl;

    }

};

class Car: virtual public Vehicle{

    protected:

    int seats;

    char air;

    public:

    Car(){

        cout << "Number of seats" << endl;

        cin >> seats ;

        cout << "Air conditioned?" << endl;

        cin >> air;

    }

    void vehicleData() const override {

        Vehicle :: vehicleData();

        cout << "Number of seats" << seats << endl;

        cout << "Air conditioned: " << air << endl;

    }

};

class Truck : virtual public Vehicle{

    protected:

    float cargo;

    int wheels;

    public:

    Truck(){

        cout << "Cargo capacity: ";

        cin>> cargo;

        cout << endl << "Number of wheels: ";

        cin >> wheels;

    }

    void vehicleData() const override{

        Vehicle :: vehicleData();

        cout << "Cargo capacity: " << cargo;

        cout << endl << "Number of wheels: " << wheels << endl;

    }

};

class PickupTruck : public Car, public Truck{

    public:

    PickupTruck(){

        cout << "Number of seats" << endl;

        cin >> seats ;

        cout << "Air conditioned?" << endl;

        cin >> air;

        cout << "Cargo capacity: ";

        cin>> cargo;

        cout << endl << "Number of wheels: ";

        cin >> wheels;

    }

    void vehicleData()const override {

        Vehicle :: vehicleData();

        Car :: vehicleData();

        Truck :: vehicleData();

    }

};

int main(){

    Vehicle \* vehiclePtr = nullptr;

    int choice;

while(choice!=-1){

    cout << endl;

    cout << "Choose the type of vehicle to create:\n";

    cout << "1. Car\n";

    cout << "2. Truck\n";

    cout << "3. Pickup Truck\n";

    cout << "-1 to exit\n";

    cout << "Enter your choice: ";

    cin >> choice;

    switch (choice) {

        case 1:

            vehiclePtr = new Car();

            break;

        case 2:

            vehiclePtr = new Truck();

            break;

        case 3:

            vehiclePtr = new PickupTruck();

            break;

        case -1:

            exit(0);

            break;

        default:

            cout << "Invalid choice!" << endl;

            break;

    }

    vehiclePtr->vehicleData();

    delete vehiclePtr;

}

    return 0;

}

**Question-3**

#include <iostream>

#include <string>

using namespace std;

class Person {

protected:

string name;

int age;

string address;

public:

Person() {

cout << "Enter name: ";

cin >> name;

cout << "Age: ";

cin >> age;

cout << "Address: ";

cin >> address;

}

virtual void display() const {

cout << "----Information-----" << endl;

cout << "Name: " << name << endl;

cout << "Age: " << age << endl;

cout << "Address: " << address << endl;

}

};

class Student : virtual public Person {

protected:

int studentID;

int GPA;

public:

Student() {

cout << "Student ID: ";

cin >> studentID;

cout << "GPA: ";

cin >> GPA;

}

void display() const override {

Person::display();

cout << "Student ID: " << studentID << endl;

cout << "GPA: " << GPA << endl;

}

};

class Teacher : virtual public Person {

protected:

int empID;

string subject;

public:

Teacher() {

cout << "Employee ID: ";

cin >> empID;

cout << "Subject: ";

cin >> subject;

}

void display() const override {

Person::display();

cout << "Employee ID: " << empID << endl;

cout << "Subject: " << subject << endl;

}

};

class TeachingAssistant : public Student, public Teacher {

public:

TeachingAssistant() {}

void display() const override {

Student::display();

cout << "Employee ID: " << empID << endl;

cout << "Subject: " << subject << endl;

}

};

int main() {

Person\* personptr = nullptr;

int choice = 0;

while (choice != -1) {

cout << endl;

cout << "Choose the attribute for object" << endl;

cout << "1. Student\n";

cout << "2. Teacher\n";

cout << "3. TeachingAssistant\n";

cout << "-1 to exit\n";

cin >> choice;

switch (choice) {

case 1:

personptr = new Student();

break;

case 2:

personptr = new Teacher();

break;

case 3:

personptr = new TeachingAssistant();

break;

case -1:

return 0; // Exit program

break;

default:

cout << "Invalid Choice" << endl;

continue;

}

personptr->display();

delete personptr;

personptr = nullptr;

}

}

return 0;

}