Lab 6 Tasks

24K-0576

Hamza Farhan

Task 1

#include <iostream>

#include <string>

using namespace std;

class Employee

{

protected:

    string name;

    float salary;

    Employee(string name, float salary) : name(name), salary(salary) {}

    void displayDetails()

    {

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

        cout << "salary: " << salary << endl;

    }

};

class Manager : public Employee

{

protected:

    float bonus;

public:

    Manager(string name, float salary, float bonus) : Employee(name, salary), bonus(bonus) {}

    void displayDetails()

    {

        Employee::displayDetails();

        cout << "bonus: " << bonus << endl;

    }

};

int main()

{

    Manager m1("hamza", 25000, 10000);

    m1.displayDetails();

}

Output



Task 2

#include <iostream>

#include <string>

using namespace std;

class Vehicle

{

protected:

    string brand;

    int speed;

public:

    Vehicle(string brand, int speed) : brand(brand), speed(speed) {}

    void displayDetails()

    {

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

        cout << "speed: " << speed << endl;

    }

};

class Car : public Vehicle

{

protected:

    int seats;

public:

    Car(string brand, int speed, int seats) : Vehicle(brand, speed), seats(seats) {}

    void displayDetails()

    {

        Vehicle::displayDetails();

        cout << "seats: " << seats << endl;

    }

};

class ElectricCar : public Car

{

    int batterylife;

public:

    ElectricCar(string brand, int speed, int seats, int bl) : Car(brand, speed, seats), batterylife(bl) {}

    void displayDetails()

    {

        Car::displayDetails();

        cout << "battery life: " << batterylife << endl;

    }

};

int main()

{

    ElectricCar e1("tesla", 250, 5, 24);

    e1.displayDetails();

}

Output

A black background with white text

AI-generated content may be incorrect.

Task 3

#include <iostream>

#include <string>

using namespace std;

class Person{

    protected:

        string name;

        int age;

    public:

        Person(string name,int age):name(name),age(age){}

        void displayDetails(){

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

            cout<<"age: "<<age<<endl;

        }

};

class Teacher: public Person{

    protected:

        string subject;

    public:

    Teacher(string name,int age,string subject):Person(name,age),subject(subject){}

    void displayDetails(){

        Person::displayDetails();

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

    }

};

class Researcher: public Person{

    protected:

        string researchArea;

    public:

    Researcher(string name,int age,string researchArea):Person(name,age),researchArea(researchArea){}

    void displayDetails(){

        Person::displayDetails();

        cout<<"research area: "<<researchArea<<endl;

    }

};

class Professor:public Researcher,public Teacher{

    protected:

        int publications;

    public:

        Professor(string name,int age,string subject,string researchArea,int publications):Teacher(name,age,subject),Researcher(name,age,researchArea),

        publications(publications){}

        void displayDetails(){

            Teacher::displayDetails();

            cout<<"research area: "<<researchArea<<endl;

            cout<<"publications: "<<publications<<endl;

        }

};

int main(){

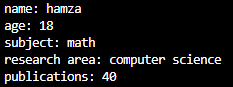
    Professor p("hamza",18,"math","computer science",40);

    p.displayDetails();

    return 0;

}

Output



Task 4

#include <iostream>

#include <string>

using namespace std;

class Account{

    protected:

        int accountNumber;

        float balance;

    public:

        Account(int accountNumber,float balance): balance(balance),accountNumber(accountNumber){}

    void displayDetails(){

        cout<<"account number: "<<accountNumber<<endl;

        cout<<"balance: "<<balance<<endl;

    }

};

class SavingsAccount:public Account{

    protected:

    float interestRate;

    public:

    SavingsAccount(int accNum,float bal,float intRate):Account(accNum,bal),interestRate(intRate){}

    void displayDetails(){

        Account::displayDetails();

        cout<<"interest rate: "<<interestRate<<"%"<<endl;

    }

};

class CheckingAccount:public Account{

    protected:

    float overdraftLimit;

    public:

    CheckingAccount(int accNum,float bal,float odLimit):Account(accNum,bal),overdraftLimit(odLimit){}

    void displayDetails(){

        Account::displayDetails();

        cout<<"overdraft limit: "<<overdraftLimit<<endl;

    }

};

int main(){

    CheckingAccount c(2215200,56000.0,6000);

    SavingsAccount s(576,60000,2.5);

    c.displayDetails();

    s.displayDetails();

    return 0;

}

Output

A screen shot of numbers

AI-generated content may be incorrect.

Task 5

#include <iostream>

#include <string>

using namespace std;

class Device{

    protected:

    bool status;

    int deviceID;

    public:

    Device(int id,bool stat):status(status),deviceID(id){}

    void displayDetails(){

        cout<<"device id: "<<deviceID<<endl;

        cout<<"status: ";

        if(status){cout<<"enable"<<endl;}else{cout<<"disable"<<endl;}

    }

};

class SmartPhone:public Device{

    protected:

    float screenSize;

    public:

    SmartPhone(int id,bool stat,float size):screenSize(size),Device(id,stat){}

    void displayDetails(){

        Device::displayDetails();

        cout<<"screensize: "<<screenSize<<endl;

    }

};

class SmartWatch:public Device{

    protected:

    bool heartRateMonitor;

    public:

    SmartWatch(int id,bool stat,bool heartRate):heartRateMonitor(heartRate),Device(id,stat){}

    void displayDetails(){

        Device::displayDetails();

        cout<<"heart rate monitor: ";

        if(heartRateMonitor){cout<<"on"<<endl;}else{cout<<"off"<<endl;}

    }

};

class SmartWearable:public SmartPhone,public SmartWatch{

    protected:

    int stepCounter;

    public:

        SmartWearable(int id,bool stat,float size,bool heartRate,int stepC):SmartPhone(id,stat,size),SmartWatch(id,stat,heartRate),

        stepCounter(stepC){}

        void displayDetails(){

            SmartWatch::displayDetails();

            cout<<"screen size: "<<screenSize<<endl;

            cout<<"step counter: "<<stepCounter<<endl;

        }

};

int main(){

    SmartWearable s(2211,true,4.4,true,5500);

    s.displayDetails();

}

Output

