OOP Lab 9 Tasks

24K-0576

Hamza Farhan

Task 1

#include <iostream>

#include <string>

using namespace std;

class Vehicle {

protected:

    string model;

    float rate;

public:

    Vehicle(string m, float r): model(m), rate(r) {}

    virtual float getDailyRate() = 0;

    virtual void displayDetail() = 0;

};

class Car : public Vehicle {

public:

    Car(string m, float r) : Vehicle(m, r) {}

    float getDailyRate() override {

        return rate;

    }

    void displayDetail() override {

        cout << "Car Model: " << model << endl;

        cout << "Daily Rate: $" << rate << endl;

    }

};

class Bike : public Vehicle {

public:

    Bike(string m, float r) : Vehicle(m, r) {}

    float getDailyRate() override {

        return rate;

    }

    void displayDetail() override {

        cout << "Bike Model: " << model << endl;

        cout << "Daily Rate: $" << rate << endl;

    }

};

int main() {

    Car myCar("Toyota Corolla", 50.0);

    Bike b("Yamaha YBR",20);

    myCar.displayDetail();

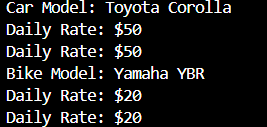
    cout << "Daily Rate: $" << myCar.getDailyRate() << endl;

    b.displayDetail();

    cout << "Daily Rate: $" << b.getDailyRate() << endl;

}

Output



Task 2

#include <iostream>

#include <string>

using namespace std;

class SmartDevice{

    public:

    SmartDevice() {

    }

    virtual void turnOn()=0;

    virtual void turnOff()=0;

    virtual bool getStatus()=0;

};

class LightBulb:public SmartDevice{

    protected:

    bool isOn;

    int brightness;

    public:

    LightBulb(int t):brightness(t){isOn=false;}

    void turnOn()override{

        isOn=true;

    }

    void turnOff()override{

        isOn=false;

    }

    bool getStatus()override{

        return isOn;

    }

};

class Thermostat:public SmartDevice{

    protected:

    bool isOn;

    double temperature;

    public:

    Thermostat(double t):temperature(t){isOn=false;}

    void turnOn()override{

        isOn=true;

    }

    void turnOff()override{

        isOn=false;

    }

    bool getStatus()override{

        return isOn;

    }

};

int main(){

    Thermostat t(47);

    LightBulb lb(70);

    t.turnOn();

    cout<<"status: "<<t.getStatus()<<endl;

    t.turnOff();

    cout<<"status: "<<t.getStatus()<<endl;

    lb.turnOn();

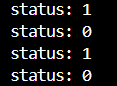
    cout<<"status: "<<lb.getStatus()<<endl;

    lb.turnOff();

    cout<<"status: "<<lb.getStatus()<<endl;

}

Output



Task 3

Book.h:

#ifndef BOOK

#define BOOK

#include <string>

using namespace std;

class Book{

    protected:

    string title,isbn,author;

    public:

    Book(string n="",string a="",string i=""):title(n),isbn(i),author(a){}

    string getTitle(){return title;}

    string getISBN(){return isbn;}

    string getAuthor(){return author;}

};

#endif

task3.cpp:

#include "Book.h"

#include <iostream>

#include <string>

using namespace std;

int main(){

    Book h("harry potter", "jk rowling","AJ5544");

    cout<<"title: "<<h.getTitle()<<"\nauthor: "<<h.getAuthor()<<"\nisbn: "<<h.getISBN()<<endl;

}

Output



Task 4

#include <iostream>

#include <string>

using namespace std;

class PaymentMethod{

    protected:

    public:

    virtual void processPayment(double amount)=0;

};

class CreditCard:public PaymentMethod{

    private:

    string cardNumber;

    public:

    CreditCard(string cn):cardNumber(cn){}

    void processPayment(double amount)override{

        cout<<"your amount "<< amount<<" is being processed\n";

    }

};

class DigitalWallet:public PaymentMethod{

    private:

    double balance;

    public:

    DigitalWallet(double b):balance(b){}

    void processPayment(double amount)override{

        if(amount>balance){

            cout<<"balance not enough\n";

        }else{

            cout<<"amount "<<amount<<" deducted from your account\nremaining balance: "<<balance-amount;

            balance-=amount;

        }

    }

};

int main(){

    CreditCard c("1221HJK2112");

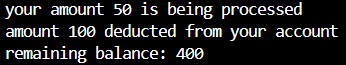
    DigitalWallet d(500.00);

    c.processPayment(50.00);

    d.processPayment(100);

}

Output



Task 5

#include <iostream>

#include <string>

using namespace std;

class Activity{

    protected:

    public:

    virtual void calculateCaloriesBurned()=0;

};

class Running:public Activity{

    private:

    double distance,time;

    public:

    Running(double d, double t) : distance(d), time(t) {}

    void calculateCaloriesBurned()override{

        cout<<"enter weight: ";

        float w;

        cin>>w;

        cout<<"calories burned: "<<time\*distance\*w<<endl;

    }

};

class Cycling:public Activity{

    private:

    double speed,time;

    public:

    Cycling(double s, double t) : speed(s), time(t) {}

    void calculateCaloriesBurned()override{

        cout<<"enter weight: ";

        float w;

        cin>>w;

        cout<<"calories burned: "<<time\*speed\*w<<endl;

    }

};

int main(){

    Running r(1.5,10);

    Cycling c(15,2);

    r.calculateCaloriesBurned();

    c.calculateCaloriesBurned();

}

Output

