**Mohammad Taha Anwer  
24K-3033  
Lab 9 Tasks Word File  
  
Task 1**

#include <iostream>

using namespace std;

class Vehicle {

protected:

string model;

double rate;

public:

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

virtual double getDailyRate() = 0;

virtual void displayDetails() = 0;

};

class Car : public Vehicle {

public:

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

double getDailyRate() { return rate \* 1.1; }

void displayDetails() {

cout << "Displaying car details..." << endl;

cout << "Model: " << model << endl << "Rate: " << getDailyRate() << endl;

}

};

class Bike : public Vehicle {

public:

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

double getDailyRate() { return rate; }

void displayDetails() {

cout << "Displaying Bike details..." << endl;

cout << "Model: " << model << endl << "Rate: " << getDailyRate() << endl;

}

};

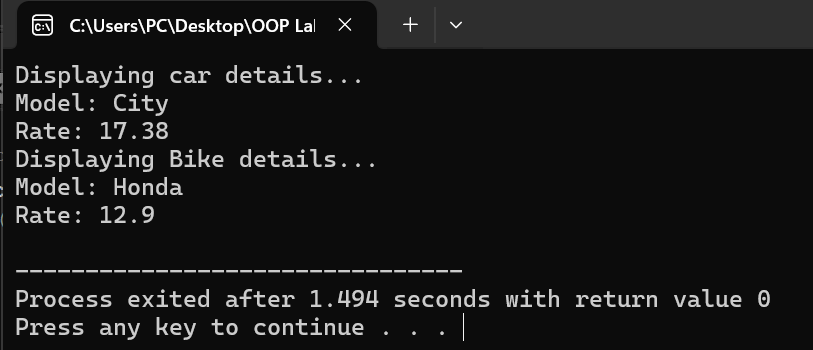
int main() {

Car c("City", 15.8);

c.displayDetails();

Bike b("Honda", 12.9);

b.displayDetails();

}  


**Task 2**

#include <iostream>

using namespace std;

class SmartDevice {

public:

virtual void turnOn() = 0;

virtual void turnOff() = 0;

virtual bool getStatus() = 0;

};

class LightBulb : public SmartDevice {

private:

bool isOn;

int brightness;

public:

LightBulb() : isOn(false), brightness(0) {}

void turnOn() {

isOn = true;

brightness = 75;

cout << "LightBulb turned on.. Brightness level set as " << brightness << endl;

}

void turnOff() {

isOn = false;

brightness = 0;

cout << "LightBulb turned off.. Brightness level set as " << brightness << endl;

}

bool getStatus() {

return isOn;

}

};

class Thermostat : public SmartDevice {

private:

bool isOn;

double temperature;

public:

Thermostat() : isOn(false), temperature(25) {}

void turnOn() {

isOn = true;

temperature = 30;

cout << "Thermostat turned on.. Temperature level is now " << temperature << endl;

}

void turnOff() {

isOn = false;

temperature = 25;

cout << "Thermostat turned off.. Temperature level is now " << temperature << endl;

}

bool getStatus() { return isOn; }

};

int main() {

LightBulb lb;

Thermostat t;

lb.turnOn();

lb.turnOff();

t.turnOn();

t.turnOff();

}  
A screenshot of a computer

AI-generated content may be incorrect.

**Task 3**

#ifndef BOOK

#define BOOK

#include <iostream>

using namespace std;

class Book {

    private:

        string title;

        string author;

        string ISBN;

    public:

        Book(string t, string a, string isbn) : title(t), author(a), ISBN(isbn) {}

        string getTitle() { return title; }

        string getAuthor() { return author; }

        string getISBN() { return ISBN; }

        void displayBook() {

            cout << endl << "Title: " << title << endl << "Author: " << author << endl << "ISBN: " << ISBN << endl << endl;

        }

};

class Library {

    private:

        Book \*\* books;

        int numBooks;

    public:

        Library() : books(nullptr), numBooks(0) {}

        void addBook(Book \* b) {

            Book\*\* tempBooks =  new Book\*[numBooks + 1];

            for(int i = 0; i < numBooks; i++) {

                tempBooks[i] =  books[i];

            }

            tempBooks[numBooks] = b;

            delete[] books;

            books = tempBooks;

            numBooks++;

            cout << "Book " << b->getTitle() << " added successfully" << endl;

        }

        int searchBook(string title) {

            for(int i = 0; i < numBooks; i++) {

                if(books[i]->getTitle() == title) {

                    cout << "Book " << title << " found at index " << i << endl;

                    return i;

                }

            }

            cout << "Book " << title << " not found..." << endl;

            return -1;

        }

        void removeBook(string title) {

            int index = searchBook(title);

            if(index == -1) {

                cout << "Cannot remove book " << title << endl;

                return;

            }

            Book\*\* tempBooks = new Book\*[numBooks - 1];

            for(int i = 0; i < index; i++) {

                tempBooks[i] = books[i];

            }

            for(int i = index + 1; i < numBooks; i++) {

                tempBooks[i - 1] = books[i];

            }

            delete books[index];

            delete[] books;

            books = tempBooks;

            numBooks--;

            cout << "Book " << title << " removed" << endl;

        }

        void displayAllBooks() {

            cout << "Displaying Library Books..." << endl;

            for (int i = 0; i < numBooks; i++) {

                books[i]->displayBook();

            }

        }

        ~Library() {

            delete[] books;

        }

};

#endif

#include "Book.h"

#include <iostream>

using namespace std;

int main() {

    Library lib;

    Book\* b1 = new Book("C++ Primer", "Lippman", "123456");

    Book\* b2 = new Book("Clean Code", "Robert C. Martin", "789101");

    Book\* b3 = new Book("Design Patterns", "Gamma", "112233");

    lib.addBook(b1);

    lib.addBook(b2);

    lib.addBook(b3);

    cout << "Initial Library:" << endl;

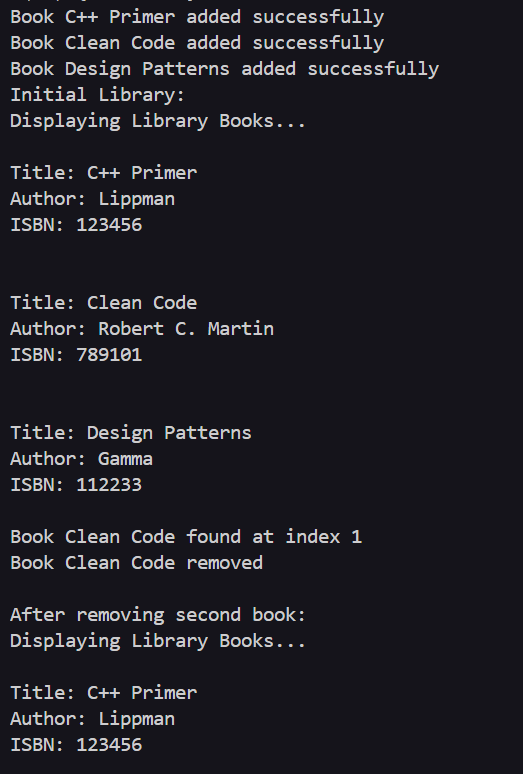
    lib.displayAllBooks();

    lib.removeBook("Clean Code");

    cout << endl << "After removing second book:" << endl;

    lib.displayAllBooks();

}

  
A screenshot of a computer

AI-generated content may be incorrect.

**Task 4**

#include <iostream>

using namespace std;

class PaymentMethod {

public:

virtual void processPayment(double amount) = 0;

};

class CreditCard : public PaymentMethod {

private:

string cardNumber;

public:

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

void processPayment(double amount) {

if(amount < 0 || amount > 5000) {

cout << "Amount out of range(0-5000)... Cannot proceed transaction" << endl;

return;

}

cout << "Transaction processed for amount " << amount << " via credit card" << endl;

}

};

class DebitCard : public PaymentMethod { // debit card = digital wallet

private:

double balance;

public:

DebitCard(double b=5000) : balance(b) {}

void processPayment(double amount) {

if(amount > balance) {

cout << "Amount exceeds the balance.. Cannot process transaction" << endl;

return;

} else if(amount < 0) {

cout << "Amount cannot be negative.. Cannot process transaction" << endl;

return;

}

balance -= amount;

cout << "Transaction processed via debit card for the amount " << amount << ", Updated Balance: " << balance << endl;

}

};

int main() {

CreditCard c("987-654-321");

DebitCard d(14000);

c.processPayment(1500);

d.processPayment(800);

c.processPayment(8000);

d.processPayment(15000);

}  
A screen shot of a computer

AI-generated content may be incorrect.

**Task 5**

#include <iostream>

using namespace std;

class Activity {

public:

virtual double calculateCalories() = 0;

};

class Running : public Activity {

private:

double distance;

double time;

public:

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

double calculateCalories() {

return distance \* 60;

}

};

class Cycling : public Activity {

private:

double speed;

double time;

public:

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

double calculateCalories() {

return speed \* time \* 40;

}

};

int main() {

Running r(50, 5);

Cycling c(30,15);

cout << "Calories burnt via running: " << r.calculateCalories() << endl;

cout << "Calories burnt via cycling: " << c.calculateCalories() << endl;

}  
A screen shot of a computer code

AI-generated content may be incorrect.