# Assignment 1

## Object Oriented Programming

**Question #1:**

You are designing a class Book for a library system. Each book has a **title**, **author**, **price, Available Quantity,** and **Category** of the book. You also need a function that calculates a **discounted price,** and **Order a Book for purchase** based on a given discount percentage. Make sure all data members should not be accessible for outer programing world. 1 constructor set all data member default values, and another should initialize it to fixed values.

Code:

Book.h

**#**include **<**iostream**>**

**#**include **<**cstring**>**

**#**include **<**iomanip**>**

using namespace std**;**

class Book **{**

private**:**

    char**\*** m\_title**;**

    char**\*** m\_author**;**

    char**\*** m\_category**;**

    double m\_price**;**

    int m\_quantity**;**

public**:**

    Book**();**

    Book**(**const char\***,** const char\***,** const char\***,** double**,** int**);**

    ~Book**();**

    void setTitle**(**const char\***);**

    void setAuthor**(**const char\***);**

    void setCategory**(**const char\***);**

    void setPrice**(**double**);**

    void setQuantity**(**int**);**

    char\* getTitle**();**

    char\* getAuthor**();**

    char\* getCategory**();**

    double getPrice**();**

    int getQuantity**();**

    void orderBook**(**int**);**

    double calculateDiscount**(**int**);**

    void printBook**();**

**};**

Book.cpp

**#**include **"**Book.h**"**

Book**::**Book**()** **{**

    m\_title **=** nullptr**;**

    m\_author **=** nullptr**;**

    m\_category **=** nullptr**;**

    m\_price **=** **0.0;**

    m\_quantity **=** **0;**

**}**

Book**::**Book**(**const char**\*** title**,** const char**\*** author**,** const char**\*** category**,** double price**,** int quantity**)** **{**

    m\_title **=** new char**[**strlen**(**title**)** **+** **1];**

    strcpy**(**m\_title**,** title**);**

    m\_author **=** new char**[**strlen**(**author**)** **+** **1];**

    strcpy**(**m\_author**,** author**);**

    m\_category **=** new char**[**strlen**(**category**)** **+** **1];**

    strcpy**(**m\_category**,** category**);**

    m\_price **=** price**;**

    m\_quantity **=** quantity**;**

**}**

Book**::~**Book**()** **{**

    delete[] m\_title**;**

    delete[] m\_author**;**

    delete[] m\_category**;**

**}**

void Book**::**setTitle**(**const char\* title**)** **{**

    delete[] m\_title**;**

    m\_title **=** new char**[**strlen**(**title**)** **+** **1];**

    strcpy**(**m\_title**,** title**);**

**}**

void Book**::**setAuthor**(**const char\* author**)** **{**

    delete[] m\_author**;**

    m\_author **=** new char**[**strlen**(**author**)** **+** **1];**

    strcpy**(**m\_author**,** author**);**

**}**

void Book**::**setCategory**(**const char\* category**)** **{**

    delete[] m\_category**;**

    m\_category **=** new char**[**strlen**(**category**)** **+** **1];**

    strcpy**(**m\_category**,** category**);**

**}**

void Book**::**setPrice**(**double price**)** **{**

    m\_price **=** price**;**

**}**

void Book**::**setQuantity**(**int quantity**)** **{**

    m\_quantity **=** quantity**;**

**}**

char\* Book**::**getTitle**()** **{**

    if **(**m\_title **==** nullptr**)** return nullptr**;**

    char**\*** temp **=** new char**[**strlen**(**m\_title**)** **+** **1];**

    strcpy**(**temp**,** m\_title**);**

    return temp**;**

**}**

char\* Book**::**getAuthor**()** **{**

    if **(**m\_author **==** nullptr**)** return nullptr**;**

    char**\*** temp **=** new char**[**strlen**(**m\_author**)** **+** **1];**

    strcpy**(**temp**,** m\_author**);**

    return temp**;**

**}**

char\* Book**::**getCategory**()** **{**

    if **(**m\_category **==** nullptr**)** return nullptr**;**

    char**\*** temp **=** new char**[**strlen**(**m\_category**)** **+** **1];**

    strcpy**(**temp**,** m\_category**);**

    return temp**;**

**}**

double Book**::**getPrice**()** **{**

    return m\_price**;**

**}**

int Book**::**getQuantity**()** **{**

    return m\_quantity**;**

**}**

double Book**::**calculateDiscount**(**int disc**)** **{**

    return m\_price **-** **(**m\_price **\*** disc **/** **100);**

**}**

void Book**::**orderBook**(**int quantity**)** **{**

    if **(**quantity **>** m\_quantity**)** **{**

        cout **<<** **"**Not enough stock available! Only **"** **<<** m\_quantity **<<** **"** books left.**"** **<<** endl**;**

        return**;**

**}**

    cout **<<** **"**Do you want to redeem discount? (y/n): **"** **<<** endl**;**

    char choice**;**

    cin **>>** choice**;**

    double total **=** **0.0;**

    if **(**choice **==** **'**y**'** **||** choice **==** **'**Y**')** **{**

        int disc**;**

        cout **<<** **"**Enter discount percentage: **";**

        cin **>>** disc**;**

        total **=** calculateDiscount**(**disc**)** **\*** quantity**;**

**}** else **{**

        total **=** m\_price **\*** quantity**;**

**}**

    m\_quantity **-=** quantity**;**

    cout **<<** **"**Order placed successfully!**"** **<<** endl**;**

    cout **<<** **"**Total amount: **"** **<<** total **<<** endl**;**

    cout **<<** **"**Remaining stock: **"** **<<** m\_quantity **<<** endl**;**

**}**

void Book**::**printBook**()** **{**

//*This function can also get values without getters*

    cout **<<** **"**Title: **"** **<<** getTitle**()** **<<** endl**;**

    cout **<<** **"**Author: **"** **<<** getAuthor**()** **<<** endl**;**

    cout **<<** **"**Category: **"** **<<** getCategory**()** **<<** endl**;**

    cout **<<** **"**Price: **"** **<<** fixed **<<** setprecision**(3)** **<<** getPrice**()** **<<** endl**;**

    cout **<<** **"**Quantity: **"** **<<** getQuantity**()** **<<** endl**;**

**}**

Void Book

**#**include **"**Book.h**"**

void book**()** **{**

    system**("**cls**");**

    Book b\_1**("**Udaas Naslen**",** **"**Abdullah Hussain**",** **"**Fiction**",** **500,** **20);**

    Book b\_2**;**

    cout **<<** **"**Enter book title**"** **<<** endl**;**

    char**\*** title **=** new char**[20];**

    cin**.**ignore**();**

    cin**.**getline**(**title**,** **20);**

    b\_2**.**setTitle**(**title**);**

    cout **<<** **"**Enter book author**"** **<<** endl**;**

    char**\*** author **=** new char**[20];**

    cin**.**getline**(**author**,** **20);**

    b\_2**.**setAuthor**(**author**);**

    cout **<<** **"**Enter book category**"** **<<** endl**;**

    char**\*** category **=** new char**[20];**

    cin**.**getline**(**category**,** **20);**

    b\_2**.**setCategory**(**category**);**

    cout **<<** **"**Enter book price**"** **<<** endl**;**

    double price**;**

    cin **>>** price**;**

    b\_2**.**setPrice**(**price**);**

    cout **<<** **"**Enter book quantity**"** **<<** endl**;**

    int quantity**;**

    cin **>>** quantity**;**

    b\_2**.**setQuantity**(**quantity**);**

    system**("**cls**");**

    cout **<<** **"**Book 1 with parameterized constructor: **"** **<<** endl**;**

    b\_1**.**printBook**();**

    cout **<<** endl **<<** **"**Book 2 with updated values: **"** **<<** endl**;**

    b\_2**.**printBook**();**

    cout **<<** **"**How many books do you want to order? **"** **<<** endl**;**

    int n**;**

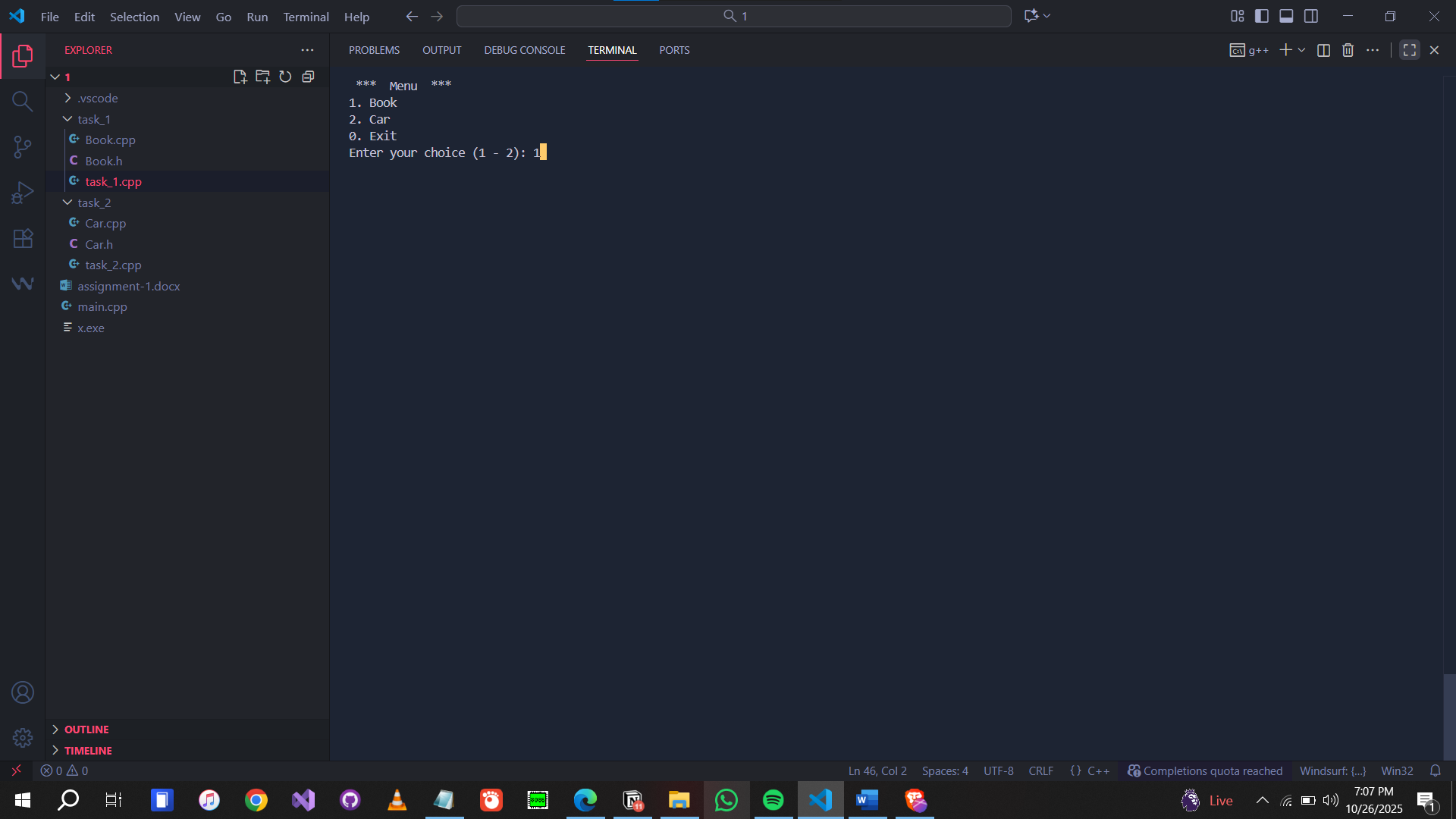
    cin **>>** n**;**

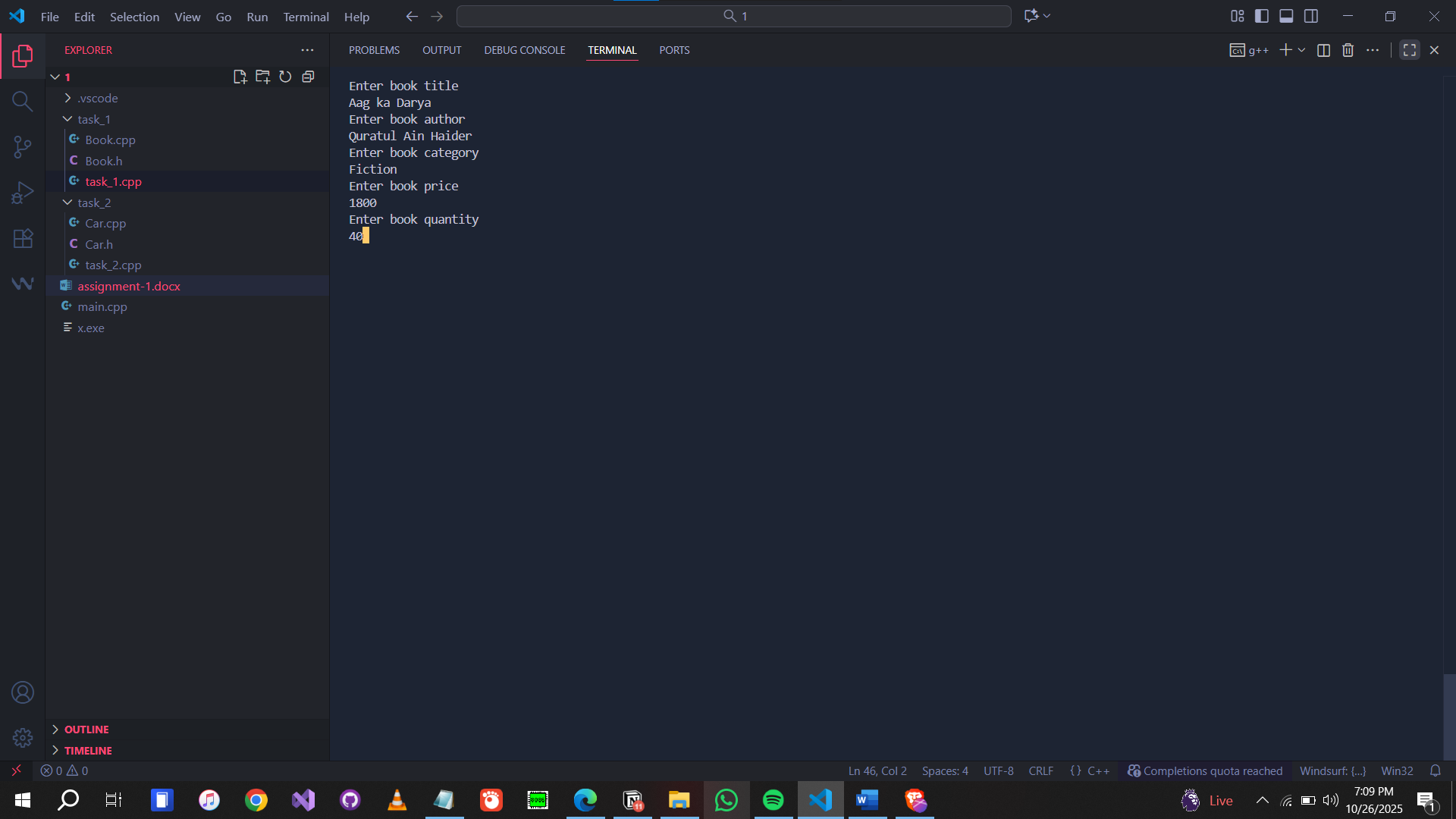
    b\_2**.**orderBook**(**n**);**

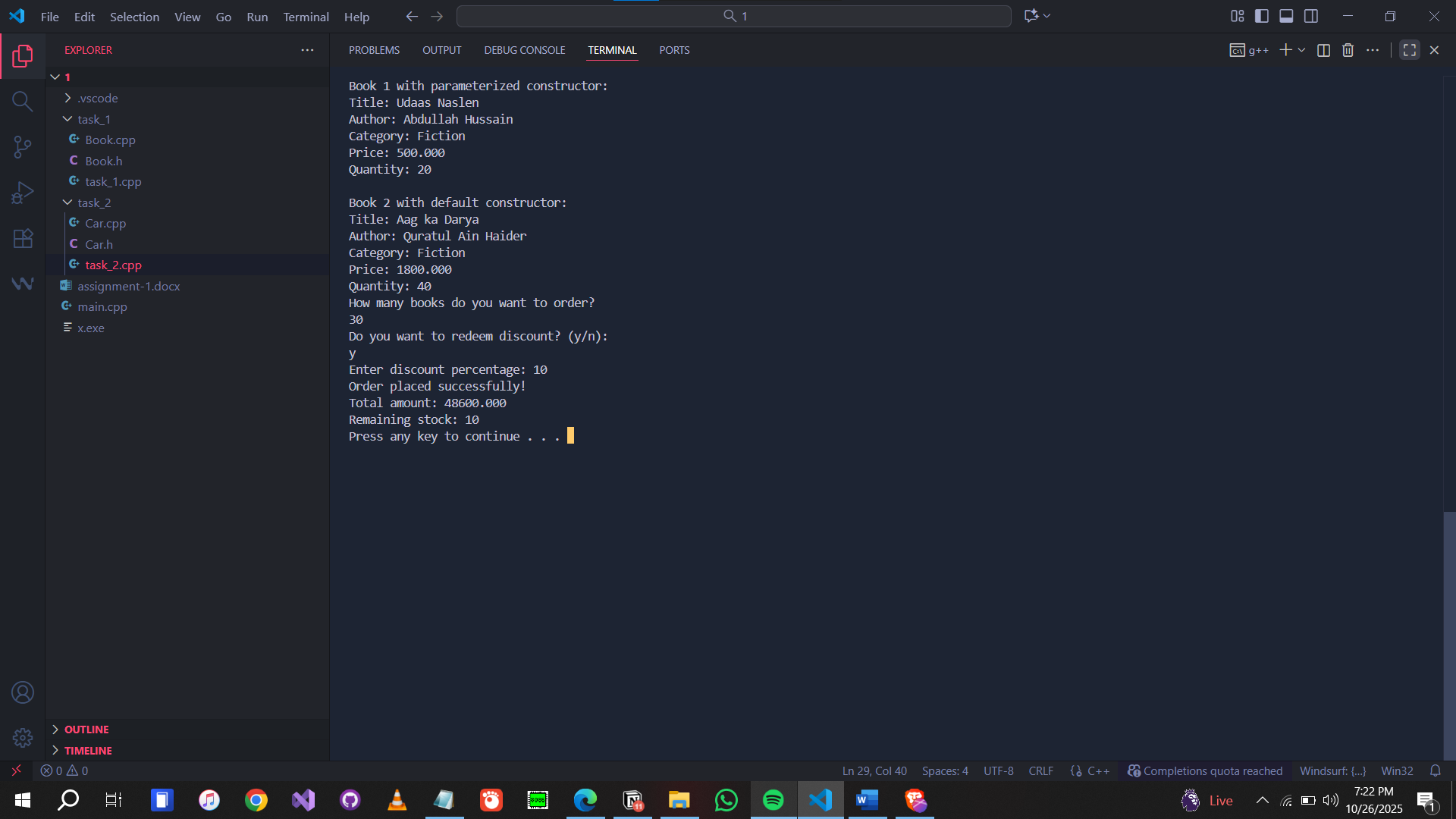
    system**("**pause**");**

**}**

Screenshot:







**Question #2:**

Write a class Car that has:

* Data members: brand, model, and price.
* A **default constructor** that sets default values ("Unknown", "N/A", 0).
* Constructor Overloads take all values
* A **function** display() that shows all details.

**Task:**

1. Create two objects: one using the default constructor and another using a **parameterized constructor**.
2. Display both cars’ details.
3. Use getter and setter function to set and get the car attributes values

Code:

Car.h

**#**include **<**iostream**>**

**#**include **<**cstring**>**

**#**include **<**iomanip**>**

using namespace std**;**

class Car **{**

private**:**

    char**\*** m\_brand**;**

    char**\*** m\_model**;**

    int m\_year**;**

    double m\_price**;**

public**:**

    Car**();**

    Car**(**const char\***,** const char\***,** int**,** double**);**

    ~Car**();**

    void setValue**(**char\***,** char\***,** int**,** double**);**

    char\* getBrand**();**

    char\* getModel**();**

    int getYear**();**

    double getPrice**();**

    void display**();**

**};**

Car.cpp

**#**include **"**Car.h**"**

Car**::**Car**()** **{**

    m\_brand **=** new char**[**strlen**("**Unknown**")** **+** **1];**

    strcpy**(**m\_brand**,** **"**Unknown**");**

    m\_model **=** new char**[**strlen**("**N/A**")** **+** **1];**

    strcpy**(**m\_model**,** **"**N/A**");**

    m\_year **=** **0;**

    m\_price **=** **0.0;**

**}**

Car**::**Car**(**const char**\*** brand**,** const char**\*** model**,** int year**,** double price**)** **{**

    m\_brand **=** new char**[**strlen**(**brand**)** **+** **1];**

    strcpy**(**m\_brand**,** brand**);**

    m\_model **=** new char**[**strlen**(**model**)** **+** **1];**

    strcpy**(**m\_model**,** model**);**

    m\_year **=** year**;**

    m\_price **=** price**;**

**}**

Car**::~**Car**()** **{**

    delete[] m\_brand**;**

    delete[] m\_model**;**

**}**

void Car**::**setValue**(**char\* brand**,** char\* model**,** int year**,** double price**)** **{**

    delete[] m\_brand**;**

    m\_brand **=** new char**[**strlen**(**brand**)** **+** **1];**

    strcpy**(**m\_brand**,** brand**);**

    delete[] m\_model**;**

    m\_model **=** new char**[**strlen**(**model**)** **+** **1];**

    strcpy**(**m\_model**,** model**);**

    m\_year **=** year**;**

    m\_price **=** price**;**

**}**

char\* Car**::**getBrand**()** **{**

    if **(**m\_brand **==** nullptr**)** return nullptr**;**

    char**\*** temp **=** new char**[**strlen**(**m\_brand**)** **+** **1];**

    strcpy**(**temp**,** m\_brand**);**

    return temp**;**

**}**

char\* Car**::**getModel**()** **{**

    if **(**m\_model **==** nullptr**)** return nullptr**;**

    char**\*** temp **=** new char**[**strlen**(**m\_model**)** **+** **1];**

    strcpy**(**temp**,** m\_model**);**

    return temp**;**

**}**

int Car**::**getYear**()** **{**

    return m\_year**;**

**}**

double Car**::**getPrice**()** **{**

    return m\_price**;**

**}**

void Car**::**display**()** **{**

//*This function can also get values without getters*

    cout **<<** **"**Brand: **"** **<<** getBrand**()** **<<** endl**;**

    cout **<<** **"**Model: **"** **<<** getModel**()** **<<** endl**;**

    cout **<<** **"**Year: **"** **<<** getYear**()** **<<** endl**;**

    cout **<<** **"**Price: Rs. **"** **<<** getPrice**()** **<<** endl**;**

**}**

Void Car

**#**include **"**Car.h**"**

void car**()** **{**

    system**("**cls**");**

    Car c\_1**("**Mercedes**",** **"**Benz**",** **2022,** **1000000);**

    Car c\_2**;**

    cout **<<** **"**Car 1 with parameterized constructor: **"** **<<** endl**;**

    c\_1**.**display**();**

    cout **<<** endl **<<** **"**Car 2 with default constructor: **"** **<<** endl**;**

    c\_2**.**display**();**

    system**("**pause**");**

    char**\*** brand **=** new char**[20];**

    char**\*** model **=** new char**[20];**

    int year **=** **0;**

    double price **=** **0;**

    cout **<<** endl **<<** **"**Enter car brand: **"** **<<** endl**;**

    cin**.**ignore**();**

    cin**.**getline**(**brand**,** **20);**

    cout **<<** **"**Enter car model: **"** **<<** endl**;**

    cin**.**getline**(**model**,** **20);**

    cout **<<** **"**Enter car year: **"** **<<** endl**;**

    cin **>>** year**;**

    cout **<<** **"**Enter car price: **"** **<<** endl**;**

    cin **>>** price**;**

    c\_2**.**setValue**(**brand**,** model**,** year**,** price**);**

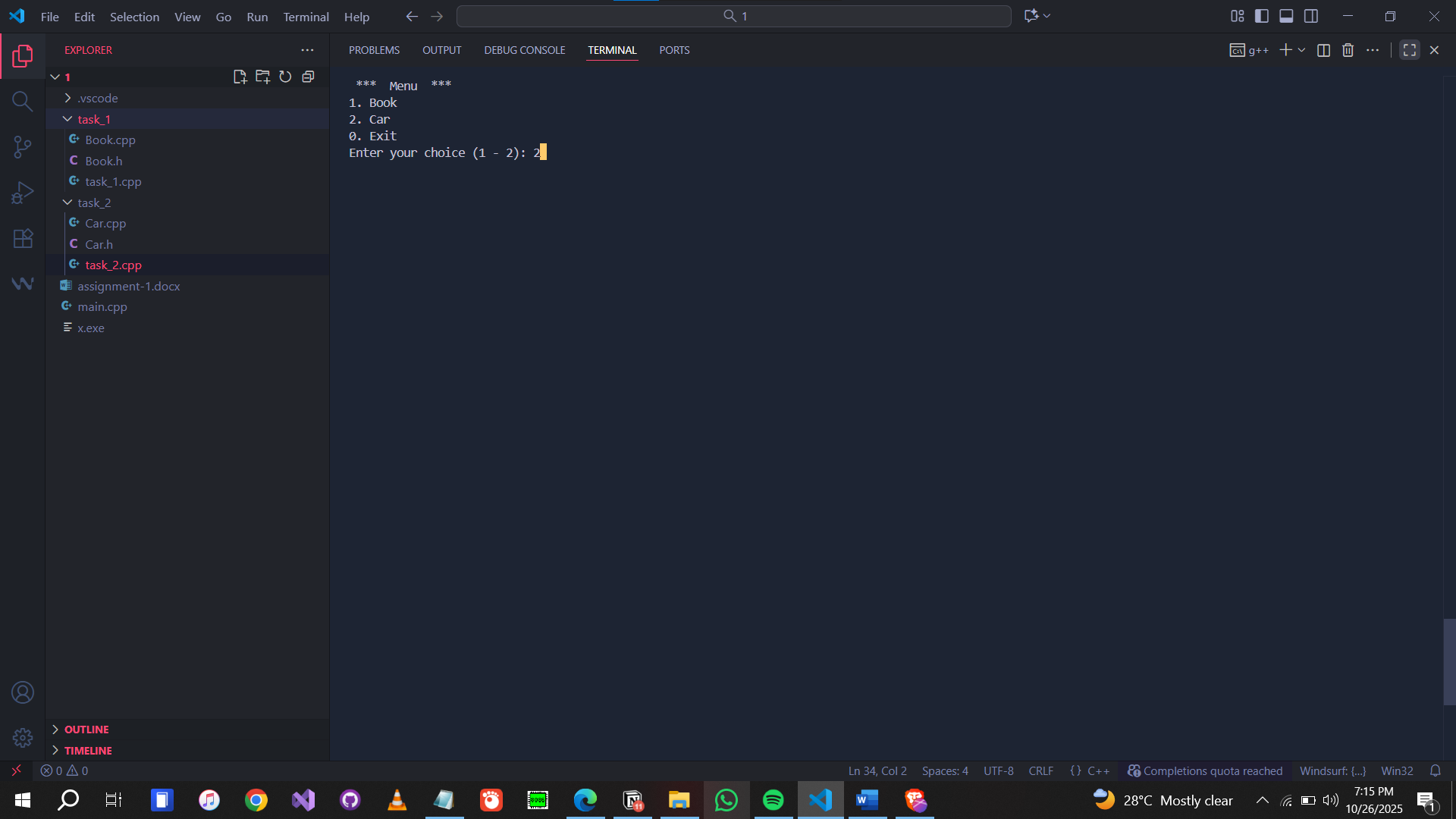
    cout **<<** endl **<<** **"**Car 2 with updated values: **"** **<<** endl**;**

    c\_2**.**display**();**

    system**("**pause**");**

**}**

Screenshot:



A computer screen with a black background

AI-generated content may be incorrect.

A computer screen with a black background

AI-generated content may be incorrect.