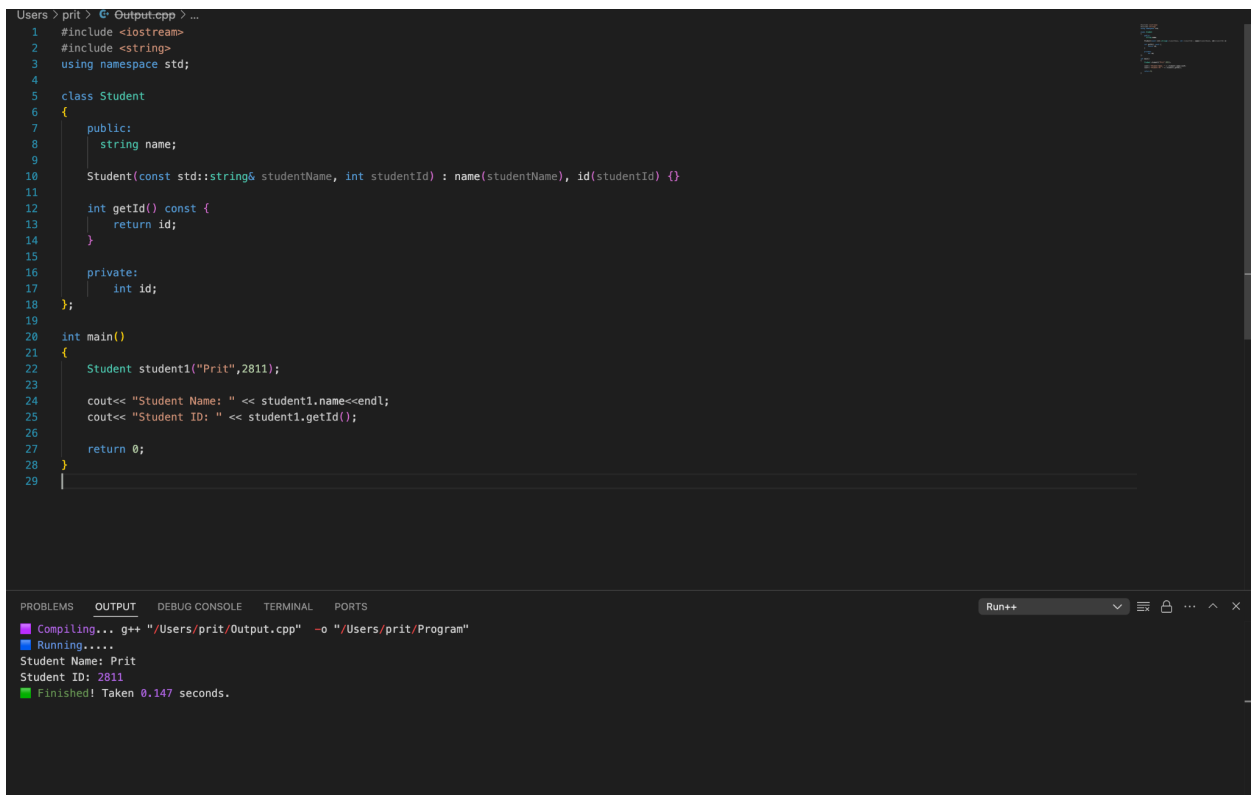# GITHUB QUESTIONS AND OUTPUT

PRIT MORI RA2411042010083

Q2) imagine you are designing a Student class. In this class, the student's name should be directly accessible (public), but the id should be private and accessible only through a getter function.

(a) How would you declare the Student class with a public name member and a private id member?
(b) Provide a code snippet that demonstrates creating a Student object, directly modifying the name, and accessing the id via its getter function.
(c) Why might a designer choose to allow direct access to some members while restricting others?

```cpp
#include <iostream>
#include <string>
using namespace std;

class Student
{
    public:
        string name;

        Student(const std::string& studentName, int studentId) : name(studentName), id(studentId) {}

        int getId() const {
            return id;
        }

    private:
        int id;
};

int main()
{
    Student student1("Prit",2811);

    cout<< "Student Name: " << student1.name<<endl;
    cout<< "Student ID: " << student1.getId();

    return 0;
}
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
Compiling... g++ "/Users/prit/Output.cpp"  -o "/Users/prit/Program"
Running.....
Student Name: Prit
Student ID: 2811
Finished! Taken 0.147 seconds.
```

Q4) Design a C++ class named Book with the following attributes:
title (string)
author (string)
isbn (string)
available (boolean)
Design a class named Library with the following attributes:
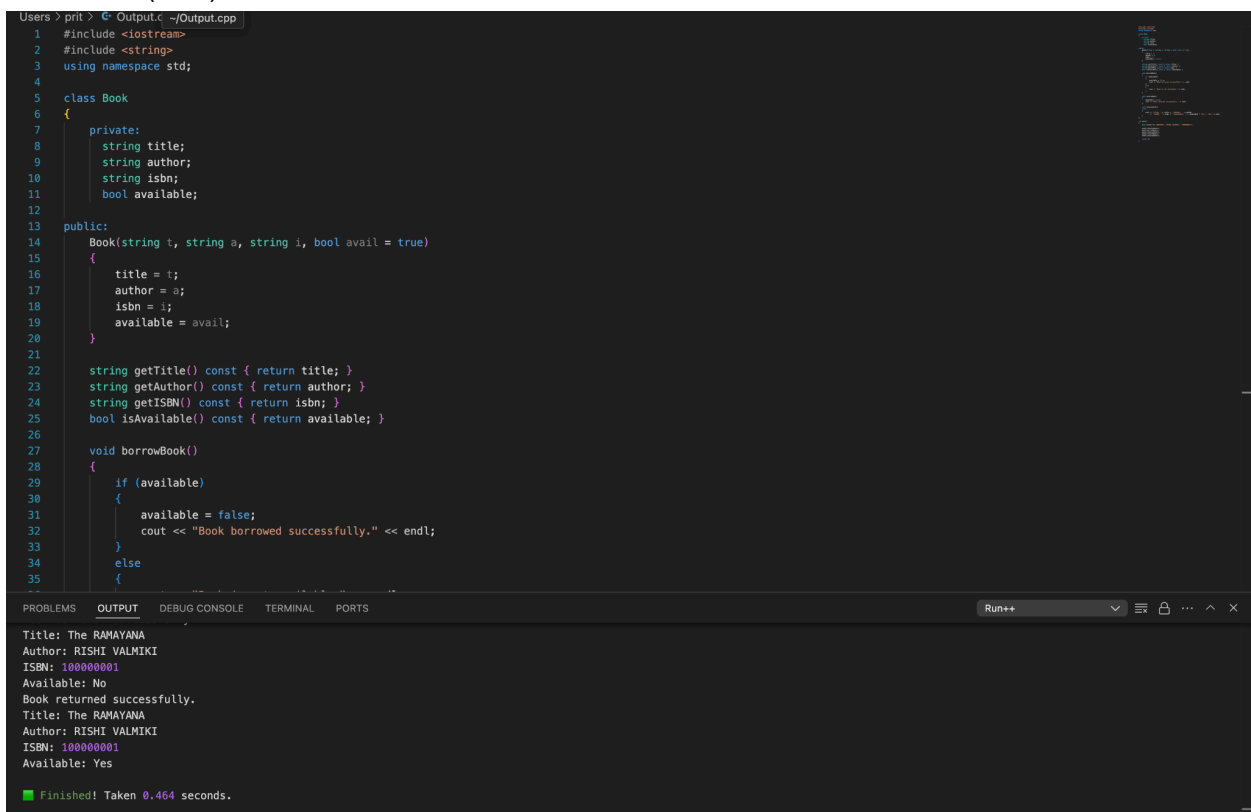books (an array or vector of Book objects)
Methods:
addBook(book): Adds a new book to the library.
searchBookByTitle(title): Searches for a book based on its title.
borrowBook(isbn): Marks a book as unavailable if it's available.
returnBook(isbn): Marks a book as available.

```cpp
Users > prit > C Output.c  ~/Output.cpp
1    #include <iostream>
2    #include <string>
3    using namespace std;
4
5    class Book
6    {
7        private:
8            string title;
9            string author;
10           string isbn;
11           bool available;
12
13       public:
14           Book(string t, string a, string i, bool avail = true)
15           {
16               title = t;
17               author = a;
18               isbn = i;
19               available = avail;
20           }
21
22           string getTitle() const { return title; }
23           string getAuthor() const { return author; }
24           string getISBN() const { return isbn; }
25           bool isAvailable() const { return available; }
26
27           void borrowBook()
28           {
29               if (available)
30               {
31                   available = false;
32                   cout << "Book borrowed successfully." << endl;
33               }
34               else
35               {
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                          Run++

Title: The RAMAYANA
Author: RISHI VALMIKI
ISBN: 100000001
Available: No
Book returned successfully.
Title: The RAMAYANA
Author: RISHI VALMIKI
ISBN: 100000001
Available: Yes

■ Finished! Taken 0.464 seconds.
```

Question:3

Design a C++ class named BankAccount with the following attributes:

accountNumber (integer)

balance (double)

accountHolderName (string)

Methods:

deposit(amount): Adds the specified amount to the balance.

withdraw(amount): Deducts the specified amount from the balance, if sufficient funds are available.

displayBalance(): Displays the current balance.

Write a program that:

Creates a BankAccount object.

Prompts the user to enter account details and initial balance.

Allows the user to perform deposit and withdrawal operations.

Displays the final balance

```cpp
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class BankAccount
6  {
7      private:
8          int accountNumber;
9          double balance;
10         string accountHolderName;
11
12     public:
13         BankAccount(int accNumber, string accHolderName, double initialBalance)
14         {
15             accountNumber = accNumber;
16             accountHolderName = accHolderName;
17             balance = initialBalance;
18         }
19
20         void deposit(double amount)
21         {
22             if (amount > 0)
23             {
24                 balance = balance + amount;
25                 cout << "Deposit successful. New balance: " << balance << endl;
26             }
27             else
28             {
29                 cout << "Invalid deposit amount. Please enter a positive value." << endl;
30             }
```

input

```
Enter Account Number: 2811
Enter Account Holder Name: Prit Mori
Enter Initial Balance: 28110000

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 1
Enter amount to deposit: 1000000
Deposit successful. New balance: 2.911e+07
```

Scenario 1: Pointers to Class Members
You're developing a vehicle management system with a class Vehicle that has several data members (for example, an integer speed and a float fuelLevel) and a member function displayStatus(). Parts of the system need to dynamically modify and display these attributes using pointers to class members.

(a) How would you declare a pointer to the speed data member and a pointer to the member function displayStatus()?
(b) Provide a code snippet that uses these pointers to set the speed of a Vehicle object and then calls displayStatus().
(c) What are some potential pitfalls when using pointers to class members, and how can they be mitigated?

```
Users > prit > C Output.cpp > ...
  1    #include <iostream>
  2    using namespace std;
  3
  4    class Vehicle
  5    {
  6        public:
  7            int speed;
  8            float fuelLevel;
  9
 10        Vehicle(int spd, float fuel) : speed(spd), fuelLevel(fuel) {}
 11
 12        void displayStatus()
 13        const
 14        {
 15            cout<<"Speed:"<< speed <<" km/h, Fuel Level: "<< fuelLevel<<" liters\n";
 16        }
 17    };
 18
 19    int main()
 20    {
 21        Vehicle car(0, 50.0f);
 22        int Vehicle::*ptrSpeed = &Vehicle::speed;
 23
 24        void (Vehicle::*ptrDisplay)() const = &Vehicle::displayStatus;
 25
 26        car.*ptrSpeed = 80;
 27        (car.*ptrDisplay)();
 28
 29        return 0;
 30    }
 31
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   Run++

■ Compiling... g++ "/Users/prit/Output.cpp" -o "/Users/prit/Program"
■ Running.....
Speed:80 km/h, Fuel Level: 50 liters

■ Finished! Taken 0.324 seconds.