**TUTORIAL-1**

**SHRIPAL DEORA**

**RA2411042010073**

**CSBS- A02**

**1ST YEAR BTECH**

**COURSE- OODP(21CSC101T)**

**FACULTY- MYTHILI.R**

**PROBLEM 1**

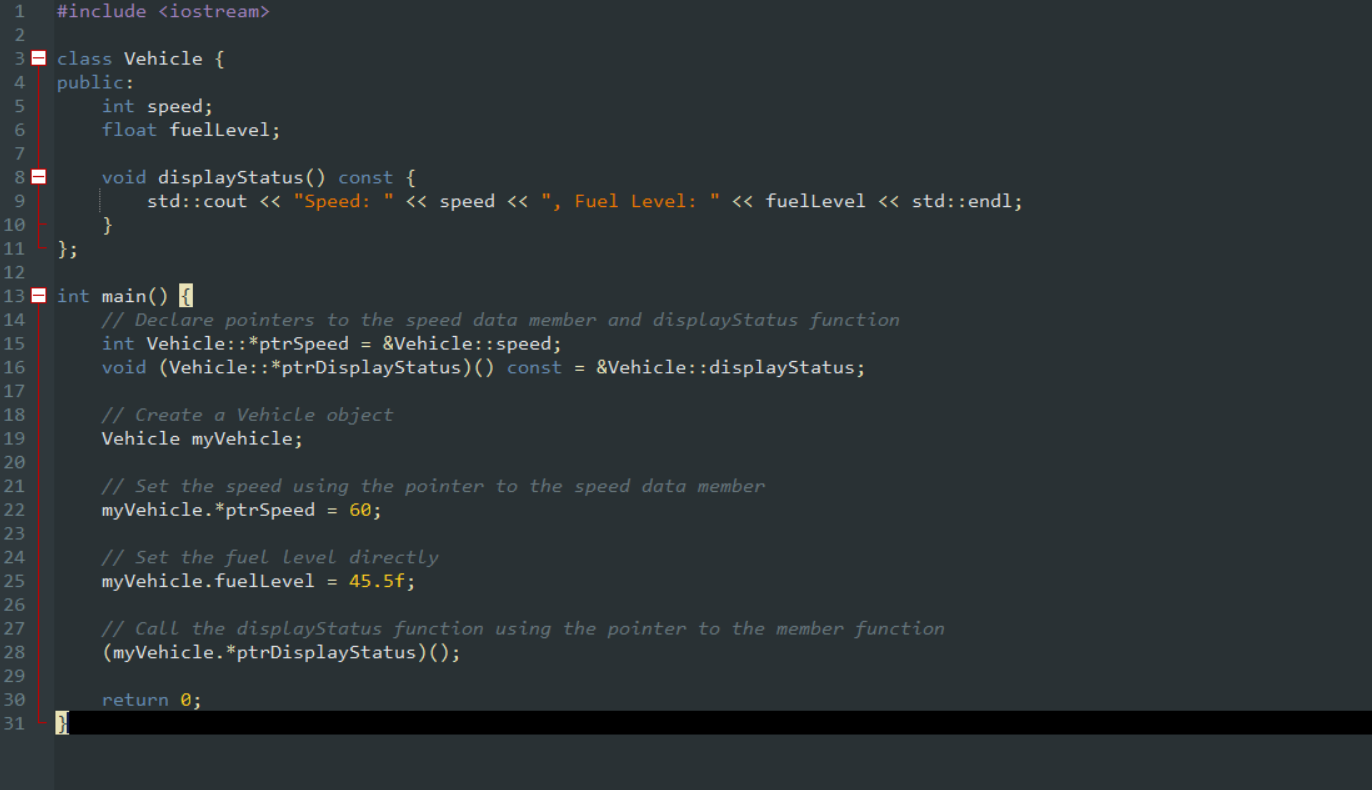
Pointers to Class Members

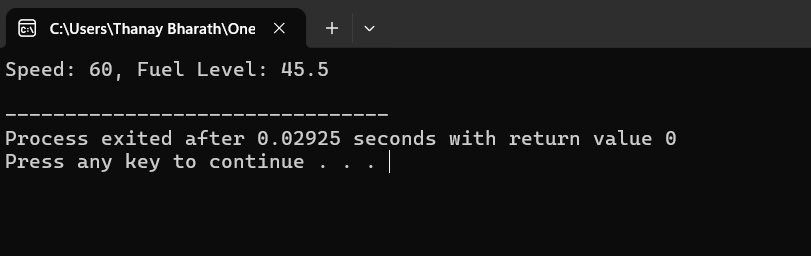
You’re developing a vehicle management system with a class Vehicle that has several data members (for example, an integer speed and a float fuelLevel) and a member function displayStatus(). Parts of the system need to dynamically modify and display these attributes using pointers to class members.

(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?

SOLUTION CODE-

OUTPUT-

**Potential Pitfalls and Mitigations**

1. **Null Pointers to Members**
   * **Issue**: If a pointer to a member is not initialized, dereferencing it may lead to undefined behavior.
   * **Mitigation**: Always initialize member pointers before use.
2. **Complex Syntax**
   * **Issue**: Using pointers to members requires .\* and ->\* operators, which are not commonly used.
   * **Mitigation**: Use typedef or using aliases to improve readability.
3. **Member Function Pointers with Different Signatures**
   * **Issue**: Calling a function pointer with an incorrect signature can cause runtime errors.
   * **Mitigation**: Ensure that function pointers match exactly, and consider using std::function or std::bind for safer function binding.

**PROBLEM-2**

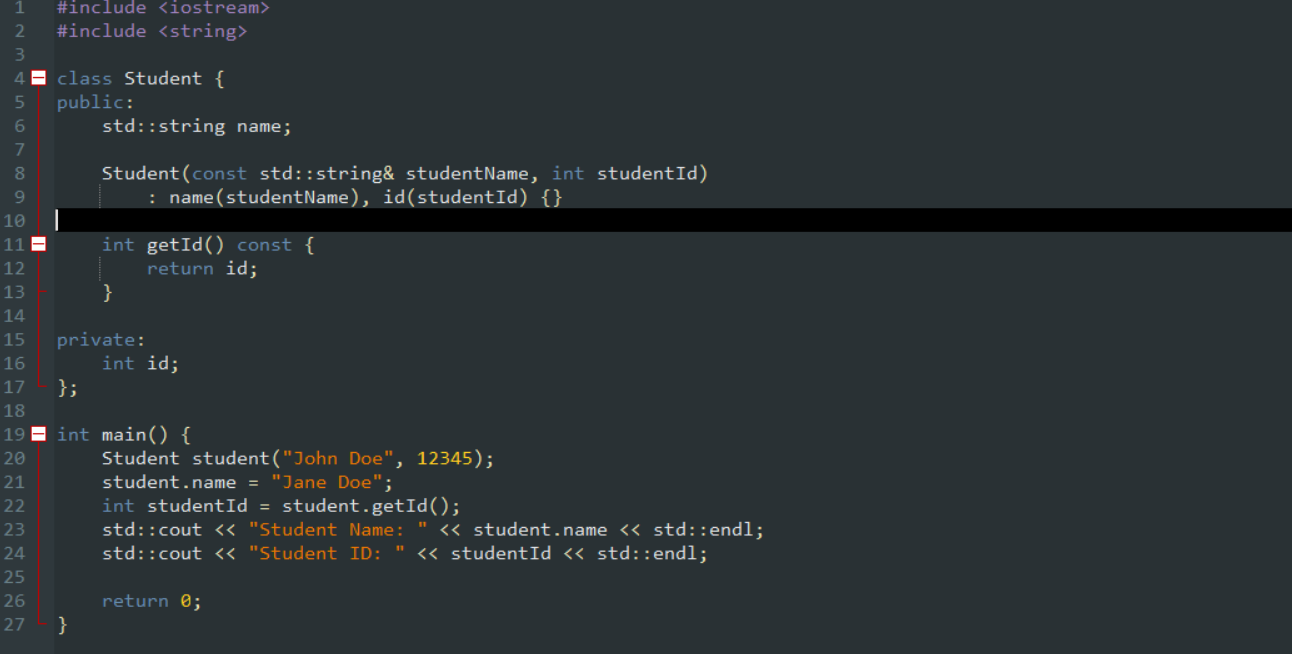
Imagine you are designing a Student class. In this class, the student’s name should be directly accessible (public), but the id should be private and accessible only through a getter function.

(a) How would you declare the Student class with a public name member and a private id member?

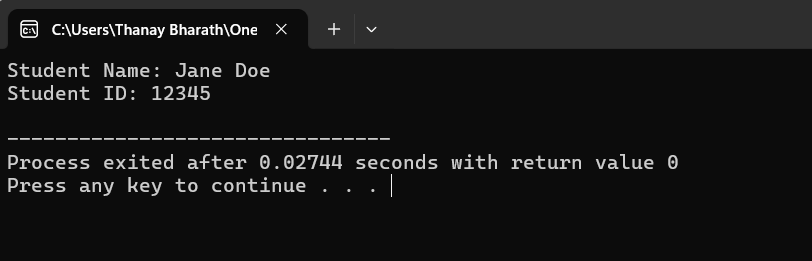
(b) Provide a code snippet that demonstrates creating a Student object, directly modifying the name, and accessing the id via its getter function.

(c) Why might a designer choose to allow direct access to some members while restricting others?

SOLUTION CODE-



OUTPUT-



**Why Allow Direct Access to Some Members While Restricting Others?**

A class designer might choose to allow direct access to some members (like name) while restricting access to others (like id) for the following reasons:

1. **Encapsulation**: By making id private, the class ensures that it cannot be modified directly from outside the class. This protects the integrity of the data and prevents unintended changes.
2. **Controlled Access**: A getter function (like getId) allows the class to control how the id is accessed. For example, the class could add validation or logging in the getter function if needed.
3. **Flexibility**: If the implementation of id changes in the future (e.g., switching from an int to a std::string), the getter function can be updated without affecting the external code that uses it.

**PROBLEM-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:

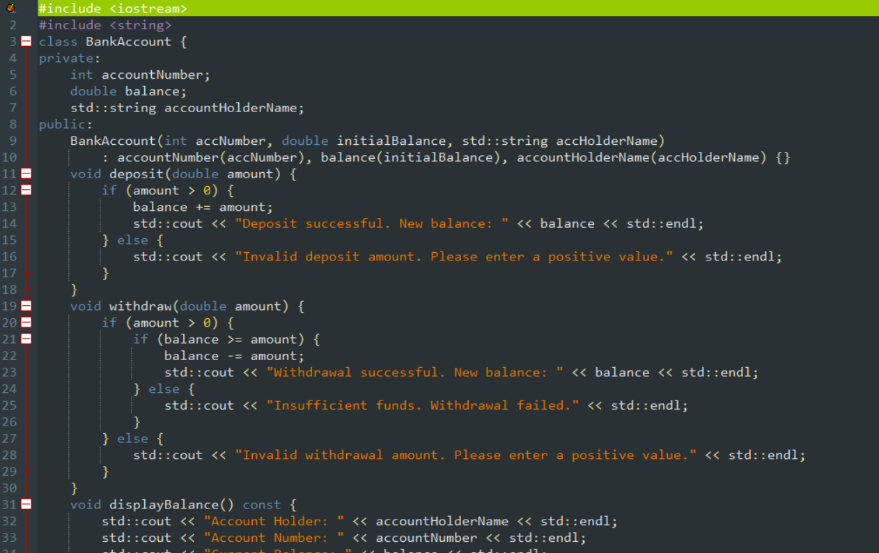
Creates a BankAccount object.

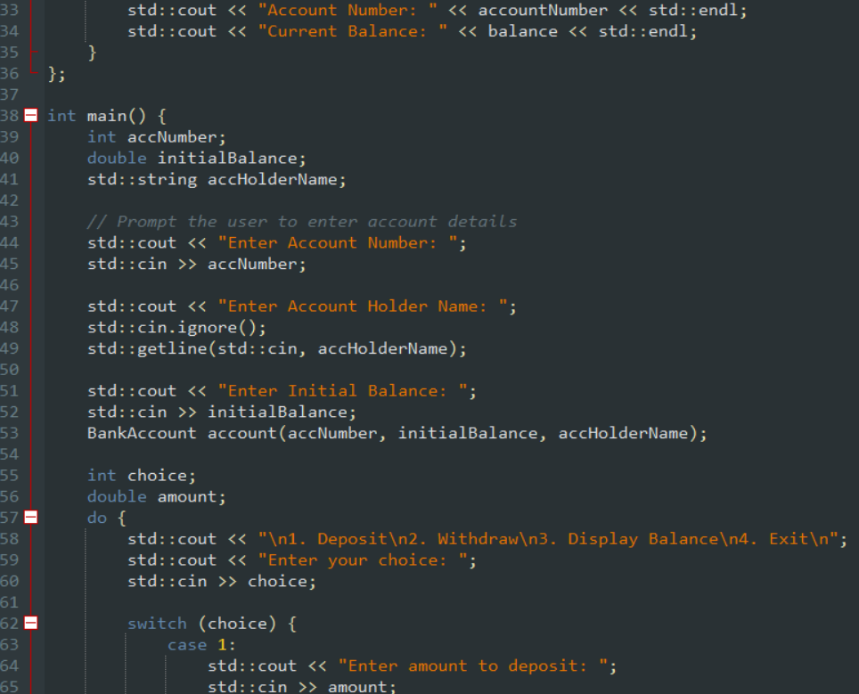
Prompts the user to enter account details and initial balance.

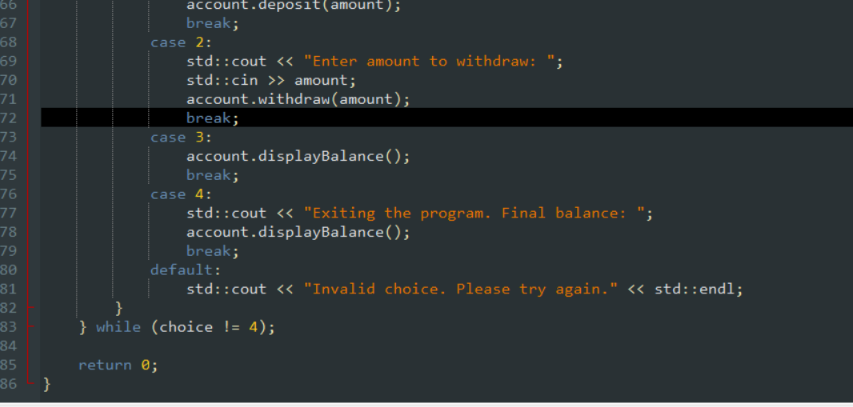
Allows the user to perform deposit and withdrawal operations.

Displays the final balance

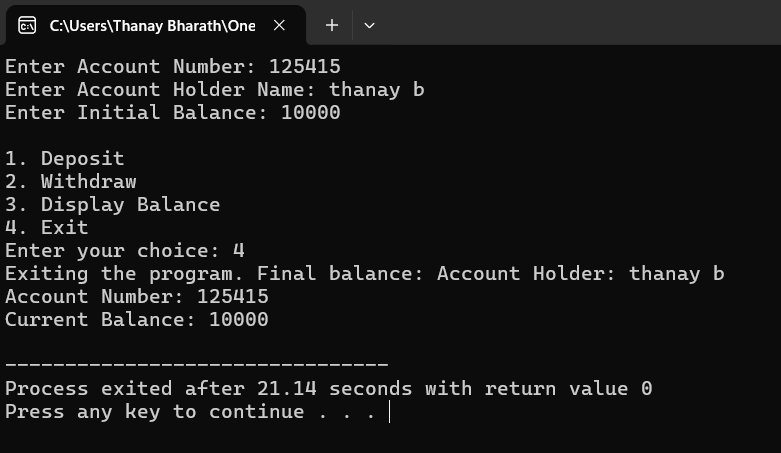
SOLUTION CODE-







OUTPUT-



**PROBLEM 4**

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)

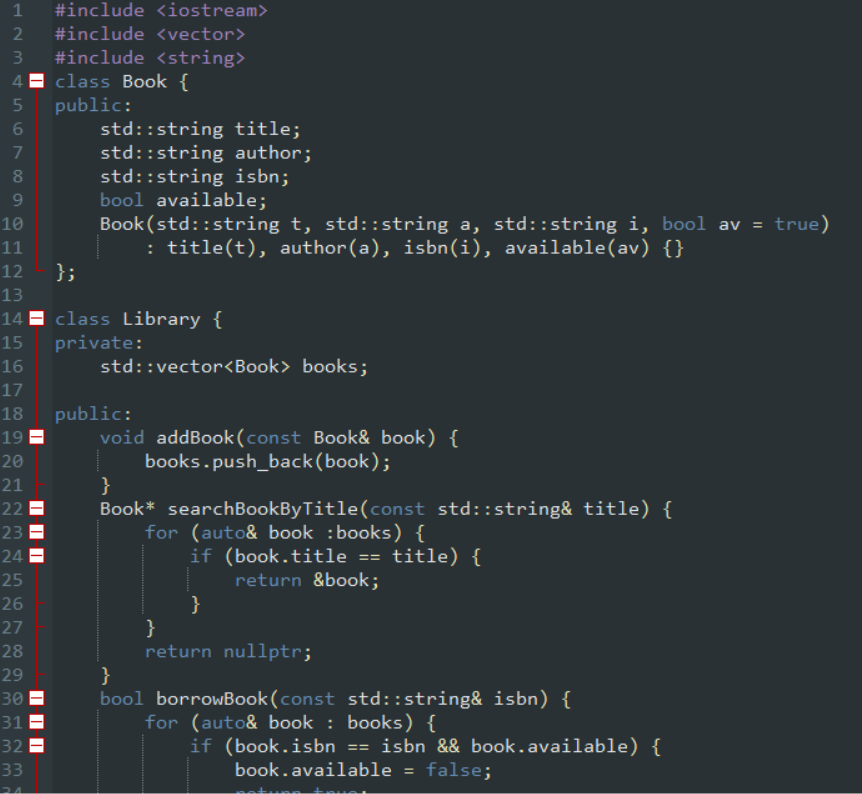
Methods:

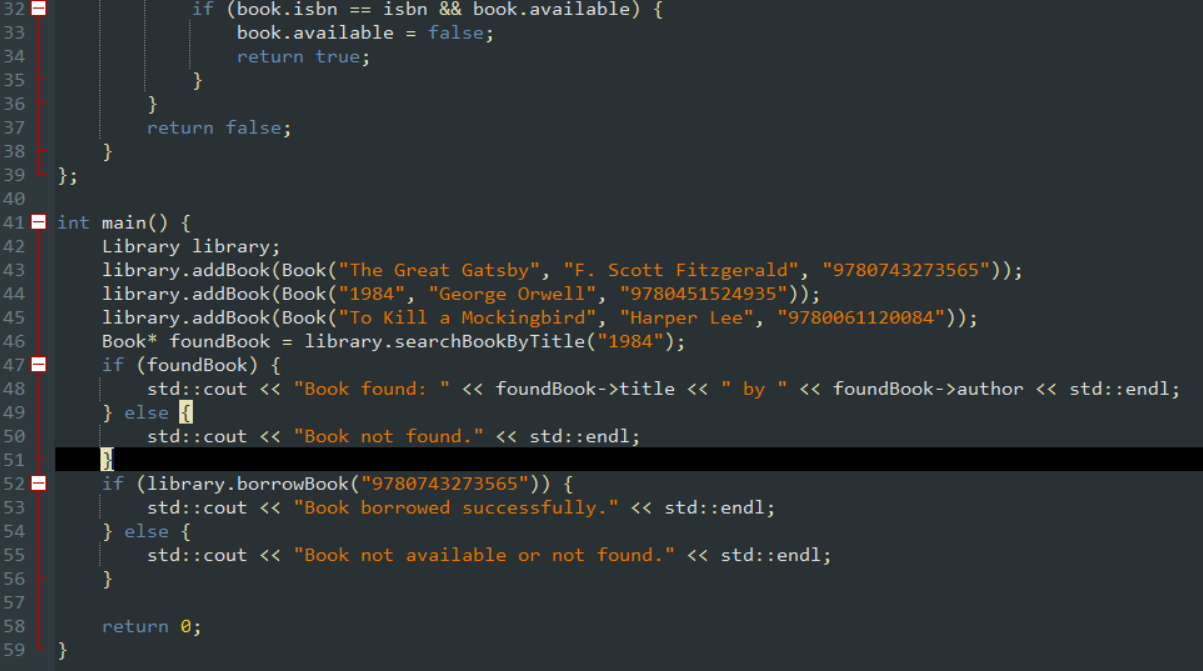
addBook(book): Adds a new book to the library.

searchBookByTitle(title): Searches for a book based on its title.

borrowBook(isbn): Marks a book as unavailable if it's available.

SOLUTION CODE-





OUTPUT-

