◇ Problem 1: Student Class with Private Data Members

Problem Statement:

Create a class `Student` that has the following private data members:

- `name` (string)
- `rollNumber` (int)
- `grade` (char)

Provide public **getter** and **setter** methods to access and modify these variables. In the `main()` function, create an object of `Student`, set the values using setters, and display them using getters.

◇ Problem 2: Bank Account with Validation

Problem Statement:

Design a class `BankAccount` with the following private data members:

- `accountNumber` (string)
- `balance` (double)

Provide setters and getters for both variables. Ensure the setter for `balance` does **not allow a negative value** to be set. Demonstrate usage in the `main()` function by trying to set both valid and invalid balance values.

◇ Problem 3: Rectangle Class with Area Calculation

Problem Statement:

Create a class `Rectangle` that has:

- Private data members: `length` and `width` (both float)
- Public setters and getters
- A public method `getArea()` that calculates and returns the area

In `main()`, create a `Rectangle` object, use setters to assign values, and print the area.

◇ Problem 4: Employee Class with Salary Privacy

Problem Statement:

Write a class `Employee` with:

- Private members: `id` (int), `name` (string), `salary` (float)
- Getters and setters
- Only allow **read access** to `salary` (i.e., no setter for salary)

In `main()`, demonstrate how salary can be accessed but not changed after initialization via constructor.

◇ Problem 5: Product Class with Price Control

Problem Statement:

Define a class `Product` with:

- Private members: `productId` (int), `productName` (string), `price` (double)

Provide:

- Getters and setters
- Validation in the `setPrice()` method to make sure the price is **greater than 0**
- In `main()`, demonstrate both successful and failed attempts to set price

---