Practice programs

C ++

Problem Statement 1

- Implement a C++ class called **BankAccount** with private data members **balance** and **accountNumber**. Create a **friend function** named **transferFunds** outside the class, which takes two **BankAccount** objects as parameters and transfers a specified amount from one account to another, updating their balances accordingly.

```
#include <iostream>
using namespace std;
class BankAccount {
private:
    float balance;
    string accountNumber;
public:
    BankAccount(float bal, string accNum) : balance(bal), accountNumber(accNum) {}
    friend void transferFunds(BankAccount &from, BankAccount &to, float amount);
    void display() {
        cout << "Account Number: " << accountNumber << ", Balance: " << balance << endl;
    }
};
void transferFunds(BankAccount &from, BankAccount &to, float amount) {
    if (amount <= from.balance) {
        from.balance -= amount;
        to.balance += amount;
        cout << "Transfer successful!" << endl;
    } else {
        cout << "Insufficient funds for transfer!" << endl;
    }
}
```

```
int main() {
    BankAccount acc1(1000.0, "12345");
    BankAccount acc2(500.0, "67890");
    acc1.display();
    acc2.display();
    transferFunds(acc1, acc2, 300.0);
    acc1.display();
    acc2.display();
    return 0;
}
```

Problem Statement 2

Create a C++ class called **Student** with private data members **name**, **rollNumber, and grade**. Implement public member functions **setData()** to set the data members, **displayData()** to display the data members, and **calculateGrade(**) to calculate the grade based on marks obtained. Ensure that name and rollNumber are private while grade is public.

```
#include <iostream>
#include <string>
using namespace std;
class Student {
private:
   string name;
   int rollNumber;
public:
   char grade;
void calculateGrade(int marks) {
       if (marks >= 90) {
          grade = 'A';
       } else if (marks >= 80) {
          grade = 'B';
       } else if (marks >= 70) {
          grade = 'C';
       } else if (marks >= 60) {
          grade = 'D';
       } else {
          grade = 'F';
       }
   }
   void setData(string n, int roll) {
          name = n;
          rollNumber = roll;
      }
      void displayData() {
          cout << "Name: " << name << ", Roll Number: " <<
  rollNumber << ", Grade: " << grade << endl;
      }
  };


int main() {
   Student s;
   s.setData("John", 101);
   s.calculateGrade(85);
   s.displayData();
   return 0;
}
```

Problem Statement 3

You are tasked with developing a student management system for a school.

The system should allow users to perform various operations such as adding new students, updating student information, deleting student records, and displaying student details.

Each student record should contain the student's name, roll number, and marks obtained in different subjects. The system should also calculate the percentage of each student based on their marks.

Additionally, the system should keep track of the total number of student records stored in the system. Implement the system using object-oriented programming concepts in C++.

all in one.cpp