NAME: SAMMY EDYSON NDERITU

REG.NO: 24/01848

Assignment – Inheritance

Question:

You are required to design a simple banking system using object -oriented programming in C++. The system should involve two classes:

1. **BankAccount** (Base Class)
   - Contains protected attributes:
     - accountHolder (string)
     - balance (double)
   - Has two public methods:
     - setDetails(string name, double amount): sets the account holder's name and initial balance.
     - displayDetails (): displays the account holder's name and balance.
2. **SavingsAccount** (Derived Class)
   - Inherits publicly from BankAccount
   - Adds one method:
     - showSavingsInfo (): prints a message displaying the savings account information using inherited attributes.

Requirements:

- Use **public inheritance** to derive SavingsAccount from BankAccount.
- In the main() function:
  - Prompt the user to enter the account holder's name and initial deposit.
  - Use the class methods to:
    - Set account details.
    - Display the account information using both displayDetails () and showSavingsInfo ().

Expected Output:

Enter account holder's
name: Jane Doe Enter
initial deposit amount:
5000 Account Holder: Jane
Doe
Balance: $5000
Savings account for Jane Doe has a balance of $5000

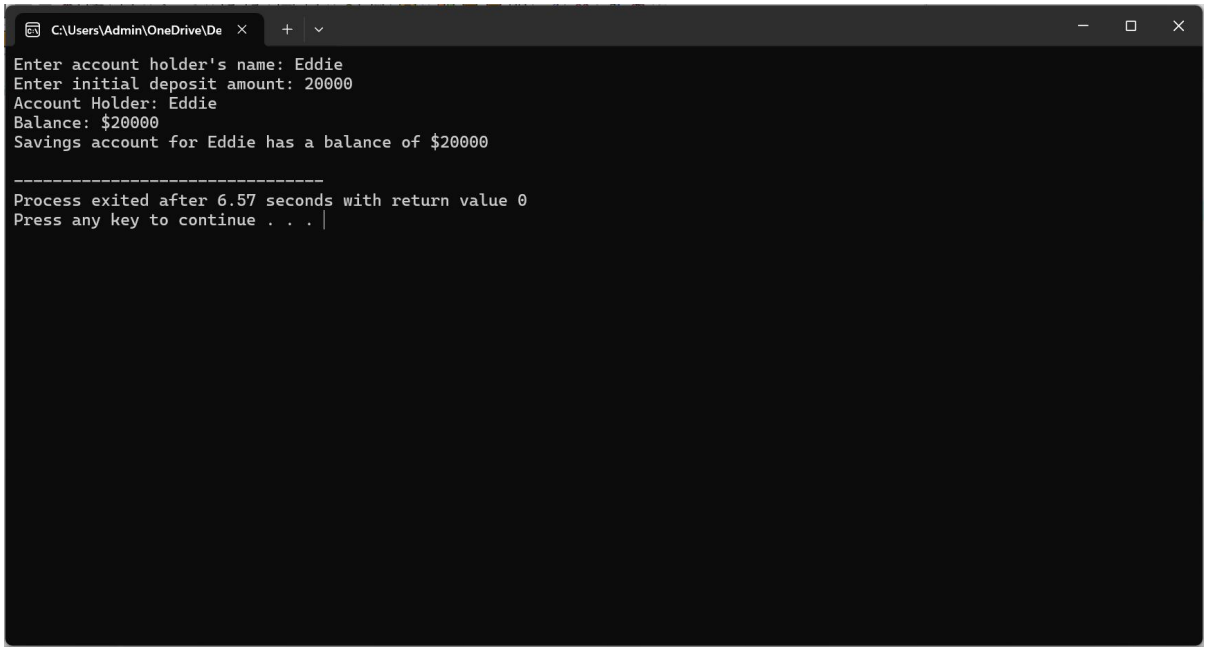
Now students input

- Add a method to calculate and display interest after 1 year (assume 5% interest rate).
- Add deposit() and withdraw() methods to theSavingsAccount class.

Part one

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

// Base class for general bank account
class BankAccount {
protected:
    string accountHolder;
    double balance;

public:
    void setDetails(string name, double amount) {
        accountHolder = name;
        balance = amount;
    }

    void displayDetails() {
        cout << "Account Holder: " << accountHolder << endl;
        cout << "Balance: $" << balance << endl;
    }
};

class SavingsAccount : public BankAccount {
public:
    void showSavingsInfo() {
        cout << "Savings account for " << accountHolder
             << " has a balance of $" << balance << endl;
    }
};

int main() {
    string name;
    double amount;

    cout << "Enter account holder's name: ";
    getline(cin, name);

    cout << "Enter initial deposit amount: ";
    cin >> amount;

    SavingsAccount myAccount;
    myAccount.setDetails(name, amount);

    myAccount.displayDetails();
    myAccount.showSavingsInfo();

    return 0;
}
```

Output:



Part two

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

// Base class for general bank account
class BankAccount {
protected:
    string accountHolder;
    double balance;

public:
    void setDetails(string name, double amount) {
        accountHolder = name;
        balance = amount;
    }

    void displayDetails() {
        cout << "\nAccount Holder: " << accountHolder << endl;
        cout << "Balance: $" << balance << endl;
    }

    void calculateInterest() {
        double interestRate = 0.05;
        double interest = balance * interestRate;
        cout << "Interest after 1 year at 5%: $" << interest << endl;
        cout << "Balance after 1 year: $" << balance + interest << endl;
    }
};


class SavingsAccount : public BankAccount {
```

```
public:
    void showSavingsInfo() {
        cout << "Savings account for " << accountHolder
             << " has a balance of $" << balance << endl;
    }

    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            cout << "Deposited $" << amount << ". New balance: $" << balance << endl;
        } else {
            cout << "Invalid deposit amount!" << endl;
        }
    }

    void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            cout << "Withdrawn $" << amount << ". New balance: $" << balance << endl;
        } else {
            cout << "Invalid or insufficient funds for withdrawal!" << endl;
        }
    }
};

int main() {
    string name;
    double initialDeposit;

    cout << "Enter account holder's name: ";
    getline(cin, name);

    cout << "Enter initial deposit amount: ";
    cin >> initialDeposit;

    SavingsAccount myAccount;
    myAccount.setDetails(name, initialDeposit);

    myAccount.displayDetails();
    myAccount.showSavingsInfo();

    double depositAmount, withdrawAmount;
    cout << "\nEnter amount to deposit: ";
    cin >> depositAmount;
    myAccount.deposit(depositAmount);

    cout << "Enter amount to withdraw: ";
    cin >> withdrawAmount;
    myAccount.withdraw(withdrawAmount);

    cout << endl;
    myAccount.calculateInterest();
```

```
    return 0;
}
```

Output