STIMULATING BASIC BANK OPERATIONS

INPUT:

```cpp
#include <iostream>

#include <string>


class BankAccount {

 private: std::string accountNumber;

 double balance;


 public:

  // Constructor

  BankAccount(const std::string & accNum, double initialBalance): accountNumber(accNum),

 balance(initialBalance) {}


 // Member function to deposit money

 void deposit(double amount) {

  balance += amount;

  std::cout << "Deposit successful. Current balance: " << balance << std::endl;

 }


 // Member function to withdraw money

 void withdraw(double amount) {

  if (amount <= balance) {

   balance -= amount;

   std::cout << "Withdrawal successful. Current balance: " << balance << std::endl;
```

```cpp
    } else {

      std::cout << "Insufficient balance. Cannot withdraw." << std::endl;

    }

  }

};


int main() {

  // Create a bank account object

  std::string sacno = "SB-123";

  double Opening_balance, deposit_amt, withdrawal_amt;

  Opening_balance = 1000;

  std::cout << "A/c. No." << sacno << " Balance: " << Opening_balance << std::endl;


  BankAccount account(sacno, 1000.0);


  // Deposit money into the account

  deposit_amt = 1500;

  std::cout << "Deposit Amount: " << deposit_amt << std::endl;

  account.deposit(deposit_amt);


  // Withdraw money from the account

  withdrawal_amt = 750;

  std::cout << "Withdrawl Amount: " << withdrawal_amt << std::endl;

  account.withdraw(withdrawal_amt);
```

```cpp
// Attempt to withdraw more money than the balance

withdrawal_amt = 1800;

std::cout << "Attempt to withdrawl Amount: " << withdrawal_amt << std::endl;

account.withdraw(withdrawal_amt);


return 0;

}
```

OUTPUT:

Output

```
/tmp/TkEqsE5Gw1.o
A/c. No.SB-123 Balance: 1000
Deposit Amount: 1500
Deposit successful. Current balance: 2500
Withdrawl Amount: 750
Withdrawal successful. Current balance: 1750
Attempt to withdrawl Amount: 1800
Insufficient balance. Cannot withdraw.
```

GITHUB LINK:

https://github.com/AlwinArulShelvan/Projects