**STIMULATING BASIC BANK OPERATIONS**

**INPUT:**

#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;

} 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);

// 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:

