ASSIGNMENT NO:- 08

**PROGRAM:**

**#include <iostream>**

**#include <stdexcept>**

**using namespace std;**

**class BankAccount {**

**private:**

**double balance;**

**public:**

**BankAccount() : balance(0.0) {}**

**void deposit(double amount) {**

**if (amount <= 0) {**

**throw invalid\_argument("Deposit amount must be positive.");**

**}**

**balance += amount;**

**cout << "Deposit successful. New balance: " << balance << endl;**

**}**

**void withdraw(double amount) {**

**if (amount <= 0) {**

**throw invalid\_argument("Withdrawal amount must be positive.");**

**}**

**if (amount > balance) {**

**throw runtime\_error("Insufficient funds. Cannot withdraw.");**

**}**

**balance -= amount;**

**cout << "Withdrawal successful. New balance: " << balance << endl;**

**}**

**double getBalance() const {**

**return balance;**

**}**

**};**

**int main() {**

**try {**

**BankAccount account;**

**double depositAmount, withdrawalAmount;**

**cout << "Enter deposit amount: ";**

**cin >> depositAmount;**

**account.deposit(depositAmount);**

**cout << "Enter withdrawal amount: ";**

**cin >> withdrawalAmount;**

**account.withdraw(withdrawalAmount);**

**cout << "Enter withdrawal amount: ";**

**cin >> withdrawalAmount;**

**account.withdraw(withdrawalAmount);**

**} catch (const exception& e) {**

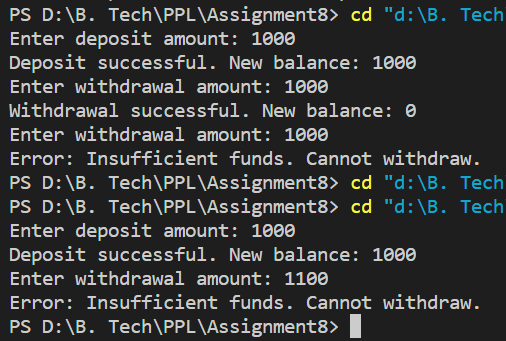
**cerr << "Error: " << e.what() << endl;**

**}**

**return 0;**

**}**

**OUTPUT:**

****