

Module 8 – Bank Account Simulation Program

The screenshot displays the Online C++ 17 Compiler IDE interface. The left pane shows the source code for `Module 8 bank_account.cpp`. The right pane shows the output of the program, which includes a summary of transactions and the current balance.

```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 using namespace std;
5 class BankAccount {
6 private:
7     double balance;
8
9 public:
10    BankAccount() { balance = 0.0; }
11
12    void deposit(double amount) {
13        balance += amount;
14        cout << "Deposited: $" << amount << endl;
15        saveTransaction("Deposit", amount);
16    }
17
18    void makePurchase(const string &item, double cost) {
19        if (cost > balance) {
20            cout << "Insufficient funds for " << item << endl;
21            saveTransaction("FAILED Purchase - " + item, cost);
22        } else {
23            balance -= cost;
24            cout << "Purchased: " << item << " for $" << cost << endl;
25            saveTransaction("Purchase - " + item, cost);
26        }
27    }
28    void displayBalance() {
29        cout << "Current Balance: $" << balance << endl;
30    }
31
32    void saveTransaction(const string &type, double amount) {
33        ofstream file("transactions.txt", ios::app);
34        if (file.is_open()) {
35            file << type << ": $" << amount << endl;
36            file.close();
37        }
38    }
39 };
40
41 // helper: print the transactions.txt to console
42 void printTransactionsFile() {
43     ifstream file("transactions.txt");
44     if (!file.is_open()) {
45         cout << "No transactions file found." << endl;
46         return;
47     }
48     cout << "\n--- transactions.txt contents ---\n";
49     string line;
50     while (getline(file, line)) {
51         cout << line << endl;
52     }
53     cout << "-----\n";
54     file.close();
55 }
56
57 int main() {
58     // Remove old transactions file for a clean run (optional)
59     // Comment out the next two lines if you want to append to previous runs:
60     std::remove("transactions.txt");
61
62     BankAccount myAccount;
63     cout << "== Bank Account Simulation == " << endl;
64
65     myAccount.deposit(200.00); // Starting deposit
66     myAccount.makePurchase("Coffee", 5.25); // Purchase 1
67     myAccount.makePurchase("Shoes", 49.99); // Purchase 2
68     myAccount.displayBalance(); // Print final balance
69
70     cout << "Transactions saved to transactions.txt" << endl;
71
72     // Print the transactions file contents so you can screenshot it
73     printTransactionsFile();
74
75     return 0;
76 }
```

The output of the program is as follows:

```
== Bank Account Simulation ==
Deposited: $200
Purchased: Coffee for $5.25
Purchased: Shoes for $49.99
Current Balance: $144.76
Transactions saved to transactions.txt

--- transactions.txt contents ---
Deposit: $200
Purchase - Coffee: $5.25
Purchase - Shoes: $49.99
-----
```

The program is compiled and executed in 1.461 sec(s).

This program simulates a bank account where the user can deposit funds, make purchases, and view the current balance.