

Perfidious ATM documentation

Liam Kelley and Shayan Amir-Kabirian

October 9, 2014

Summary

To see normal features of the ATM program, use the PIN **1234**. To see easter eggs, try the PIN **8267**.

1 Program Logic Structure

The program, instead of running as a continuous `while` loop, runs as a series of functions. There are functions for displaying the menu and functions for each of the menu options, such as Deposit or Withdrawal. These functions, when finished, then call menu again. The overall logic is as follows:

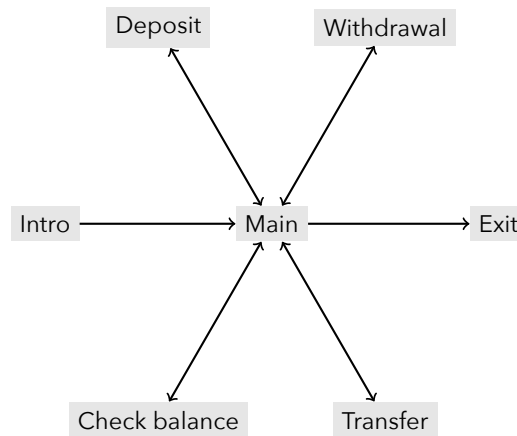


Figure 1: Function logic

The use of functions leads to potential issues with global variables, especially the balances of the two accounts. To overcome this issue, we pass the values of checking and savings along as inputs to each menu option. Therefore, calling the menu requires:

```
main(checking_account,savings_account)
```

Calling the deposit function, as another example, also requires the two accounts as inputs.

```
deposit(checking_account,savings_account)
```

By passing the account information along to each menu, we avoid having to make `checking_account` and `savings_account` global variables.

2 Additional Features and Easter Eggs

The ATM is itself fully functional, but seems a touch off . This is intentional, of course. The ATM is vaguely Germanic, with umlauts over many of the u's, and the use of Euro instead of USD. Furthermore, the company is meant to sound sketchy. Perfidious means “deceitful and untrustworthy,” and the company name itself changes over the course of the program. Each time the program says the name, the classification at the end of the company (i.e. Inc., LLC, etc.) is randomly generated using the following code:

```
1 def bank_name():
2     n = random.randint(0,18)
3     title = "GmbH","AG","AG&Co.KG" # rest left off for brevity
4     return "PerfidiousBankingGroup," + title[n]
```

The name is therefore thrown in by calling `bank_name()` in a `print()` statement.

Certain features of the ATM were made more advanced. PINs, of which there are four, are read from a file instead of hardcoded into the program. Similarly, the account information is also stored in a separate file, as is the build number. Therefore, code was implemented for reading the file and rewriting the information within, for the account information and the build number. The build number was read and iterated by one. The function that reads and outputs the build number is below as an example of text file I/O.

```
1 def build():
2     build_file = open('build.txt','r')
3     build_num = build_file.readline()
4     build_num = int(build_num.rstrip("\n"))
5     new_build = build_num + 1
6     build_file = open('build.txt', 'w')
7     build_file.write(str(new_build)+"\n")
8     return new_build
```

The documentation of each line in the function is included as comments in the program itself. As a result, the PINs, build number, and account information are all stored in `.txt` documents that the functions reads and rewrites as necessary. For account information, the account balances are read when the program is initialized and rewritten when the program exits.

The basic functionality of program is available for the default PIN, which is 1234. However, there are also three administrator PINs, which are 1248, 3333, and 8267. These correspond to Liam, Shayan, and our friend RC. Inputting any of these PINs allows for access to an admin menu with some farcical illegal options, such as money laundering.

Additionally, because our friend RC has a cherished lobster stuffed animal, there is a verification that the user is a person and not a lobster. If the user is a lobster, the program exits immediately.

The admin menu has a similar overall structure to the main menu. The options for the admin menu, which are:

1. Quick Money Laundering
2. Transfer to Cayman Account
3. Fast Embezzle €1000
4. Delete All Evidence
5. Return to Main Menu

These menu options are called in the same general way that main menu functions are called. The admin menu itself is only an option in the main menu when the global toggle `admin` is set to `True`, which occurs when an admin PIN is entered. The logic of each admin menu option is still very basic and related to normal menu options. Money laundering and transfer to Cayman account utilize the same general logic as the withdrawal function, while fast embezzle uses the same logic as deposit. Delete all evidence utilizes the same general logic as the exit functionality, and rewrites the account file to be blank (it also adds a little showmanship).

There are also two defined functions that are never called, which is a good thing because they are intentional infinite loops. The function `meow()` prints “meow” repeatedly.

```
1 def meow():  
2     meow = "meow"  
3     while meow == "meow":  
4         print(meow)
```

Meanwhile, the function `conjure_puppies()` prints “Puppies!” until the end of time, or at least until it is forcibly stopped.

```
1 def conjure_puppies():  
2     puppy_cuteness = True  
3     while puppy_cuteness:  
4         print("Puppies!")
```