

The Account Programming Assignment

1. Implement a `Transaction` class that takes an amount, a date, a currency (default “USD”—U.S. dollars), a USD conversion rate (default 1), and a description (default `None`). All of the data attributes must be private.

Provide the following read-only properties: `amount`, `date`, `currency`, `usd_conversion_rate`, `description`, and `usd` (calculated as `amount*usd_conversion_rate`).

This class can be implemented in about sixty lines including some simple *doctests*.

2. Implement an `Account` class that holds an account number, an account name, and a list of `Transactions`. The number should be a read-only property; the name should be a read-write property with an assertion to ensure that the name is at least four characters long.

The class should support the built-in `len()` function (returning the number of transactions), and should provide two calculated read-only properties: `balance` which should return the account’s balance in USD and `all_usd` which should return `True` if all the transactions are in USD and `False` otherwise.

Three other methods should be provided: `apply()` to apply (add) a transaction, `save()`, and `load()`.

The `save()` and `load()` methods should use a binary `pickle` with the filename being the account number with extension `.acc`; they should save and load the account number, the name, and all the transactions.

This class can be implemented in about ninety lines with some simple *doctests* that include saving and loading—use code such as

```
name = os.path.join(tempfile.gettempdir(), account_name)
```

to provide a suitable temporary filename, and make sure you delete the temporary file after the tests have finished

You should download the file `Account.py`; it contains the beginnings of the two classes. You are to complete them.