PRACTICE WITH CLASSES AND OBJECTS

Working with a partner/group, use the following steps to solve each of the following problems.

(a) Plan your code on the white board (either on the classroom wall or on Zoom).

(b) *Only after you have planned your code should you type your code in Python.*

(c) After you have typed your code, run multiple test cases. Does your code work? If not, how can you fix it?

1. **Account numbers**: Modify the bank account class to add an account number as a new attribute. The account number should be initialized to a random number when an account is created.

2. **Interest**: Modify the bank account class to add an interest rate as a new attribute. Then add a new method `addInterest()` that computes and adds interest to the account balance. You could simply compute the interest earned by multiplying the balance by the interest rate. How could you implement a more realistic model of interest?

3. **Sorting**: Implement the `__str__()` method to provide a text representation of a bank account object. Remember that this method should return a string containing information about the account. You may decide what to include in this string!

4. **Account history**: How could you modify the bank account class to keep track of the transaction history for the account? For example, you might add a list that tracks the account balance over time. Or, you could keep a list of transactions and account balances. Plan and implement account history tracking.

5. **Your own object**: Define a new class in Python to represent an object of your choosing. What attributes does your object require? What methods will you implement? Be creative!