1. **Technical Design Document Template**
2. **Name:** Tyler Vo
3. **Date Created:** 11/3/2024
4. **Functions Used in the Program:**
5. **Function Name:** \_\_init\_\_
   1. **Description:** Initializes the bank account with account holder's name, account number, initial balance, and interest rate.
   2. **Parameters:**
      1. holder\_name (str): Name of the account holder.
      2. acct\_number (str): Account number.
      3. balance (float, optional): Initial balance of the account.
      4. annual\_interest (float, optional): Annual interest rate.
   3. **Variables:**
      1. self.holder\_name (str): Stores the account holder's name.
      2. self.acct\_number (str): Stores the account number.
      3. self.balance (float): Stores the current balance.
      4. self.annual\_interest (float): Stores the annual interest rate.
   4. **Logical Steps:**
      1. Assigns the provided values to instance attributes.
   5. **Returns:** None
6. **Function Name**: update\_interest\_rate
   1. **Description:** Updates the interest rate for the account if within a valid range (0-1).
   2. **Parameters:**
      1. new\_interest\_rate (float): New interest rate to be applied.
   3. **Variables:**
      1. self.annual\_interest (float): Updated with the new interest rate if valid.
   4. **Logical Steps:**
      1. Checks if new\_interest\_rate is between 0 and 1.
      2. Updates self.annual\_interest if valid; otherwise, prints an error.
   5. **Returns:** None
7. **Function Name:** deposit\_funds
   1. **Description:** Deposits a specified amount into the account if positive.
   2. **Parameters:**
      1. None
   3. **Variables:**
      1. self.balance (float): Updated to reflect the deposit.
   4. **Logical Steps:**
      1. Checks if amount is positive.
      2. Adds amount to self.balance; otherwise, prints an error.
   5. **Returns:** None
8. **Function Name:** withdraw\_funds
   1. **Description:** Withdraws the amount from the account if it doesn’t exceed the balance.
   2. **Parameters:**
      1. amount (float): Amount to withdraw from the account.
   3. **Variables:**
      1. self.balance (float): Updated to reflect the withdrawal.
   4. **Logical Steps:**
      1. Checks if amount is positive and does not exceed self.balance.
      2. Subtracts amount from self.balance; otherwise, prints an error.
   5. **Returns:** None
9. **Function Name:** get\_current\_balance
   1. **Description:** Returns the current balance of the account.
   2. **Parameters:** None
   3. **Variables:**
      1. self.balance (float): Represents the current balance.
   4. **Logical Steps:**
      1. Returns the value of self.balance.
   5. **Returns:** float - the current balance of the account.
10. **Function Name:** calculate\_daily\_interest
    1. **Description:** Calculates interest over a specified number of days based on the daily interest rate.
    2. **Parameters:**
       1. days (int): Number of days to calculate interest for.
    3. **Variables:**
       1. daily\_interest\_rate (float): Daily rate derived from annual interest.
       2. interest\_earned (float): Calculated interest for the given period.
    4. **Logical Steps:**
       1. Checks if days is positive.
       2. Calculates daily\_interest\_rate by dividing self.annual\_interest by 365.
       3. Calculates interest\_earned using self.balance, daily\_interest\_rate, and days.
       4. Returns the interest\_earned amount.
    5. **Returns:** float - the calculated interest amount.
11. **Function Name:** \_\_str\_\_
    1. **Description:** Returns a formatted string representing the account holder, account number, balance, and interest rate.
    2. **Parameters:** None
    3. **Variables:**
       1. self.holder\_name,
       2. self.acct\_number,
       3. self.balance,
       4. self.annual\_interest
    4. **Logical Steps:**
       1. Formats and returns a string with account details.
    5. **Returns:** str - a string with the account details.
12. **Function Name:** test\_bank\_acct
    1. **Description:** Demonstrates the functionality of the BankAcct class by performing account operations like updating the interest rate, depositing, withdrawing, and calculating interest.
    2. **Parameters:** None
    3. **Variables:**
       1. initial\_balance (float): Initial balance for the test account.
       2. initial\_interest\_rate (float): Initial interest rate for the test account.
       3. updated\_interest\_rate (float): New interest rate to apply.
       4. deposit\_amount (float): Amount to deposit into the account.
       5. withdrawal\_amount (float): Amount to withdraw from the account.
       6. days (int): Number of days for which interest is calculated.
       7. interest (float): Holds the calculated interest for the period.
    4. **Logical Steps:**
       1. Creates a BankAcct instance with initial balance and interest rate.
       2. Displays account details.
       3. Updates interest rate and displays updated account details.
       4. Deposits an amount and displays the balance.
       5. Withdraws an amount and displays the balance.
       6. Calculates and displays interest over a specified period.
    5. **Returns:** None
13. **Logical Steps:**
    1. The test\_bank\_acct function initializes a BankAcct object.
    2. It displays account information.
    3. The interest rate is updated, and the details are printed.
    4. A deposit is made, and the balance is displayed.
    5. A withdrawal is made, and the balance is displayed.
    6. Interest is calculated for a set number of days and displayed.
14. **Link to your repository:** https://github.com/AsianInvasion00/COP2373