

## Date Testing:

| <u>Test Case #</u> | <u>Requirement</u>  | <u>Test description and Input Data</u>   | <u>Expected Result/Output</u>  |
|--------------------|---|--|--|
| 1                  | Method should not accept any date after the current date. | <ul style="list-style-type: none"><li>• Create an instance of Date with valid day and month but with year &gt; 2022</li><li>• test data: "11/30/2024"</li></ul>  | false <ul style="list-style-type: none"><li>• Date of birth invalid.</li></ul> |
| 2                  | Method should not accept the date that is today.          | <ul style="list-style-type: none"><li>• Create an instance of Date with current day and month (*written on 3/22/22)</li><li>• test data: "3/22/2022"</li></ul>   | false <ul style="list-style-type: none"><li>• Date of birth invalid.</li></ul> |
| 3                  | Valid range for month should be 1-12                      | <ul style="list-style-type: none"><li>• Create an instance of Date with valid day and year, but with month &lt; 1 or &gt; 12</li><li>• test data: "0/4/1984"</li></ul>   | false <ul style="list-style-type: none"><li>• Date of birth invalid.</li></ul> |
| 4                  | Valid range for day should be month-specific              | <ul style="list-style-type: none"><li>• Create an instance of Date with valid month and year but with day &lt; 1 or &gt; 30</li><li>• Create second instance of Date with valid month and year but with day &lt; 1 or day &gt; 31</li><li>• test data: "9/31/2000"<br/>"1/32/1999"<br/>"9/0/2000"<br/>"1/0/1999"</li></ul> | false <ul style="list-style-type: none"><li>• Date of birth invalid.</li></ul> |
| 5                  | Number of days in non-leap year February should be 28     | <ul style="list-style-type: none"><li>• Create an instance of Date with date February 29th on non-leap year</li><li>• test data: "2/29/2011"</li></ul>   | false <ul style="list-style-type: none"><li>• Date of birth invalid.</li></ul> |
| 6                  | Correct date, leap-year date                              | <ul style="list-style-type: none"><li>• Create an instance of Date with valid date.</li><li>• test data: "3/15/1995"<br/>"2/29/2012"</li></ul>   | true <ul style="list-style-type: none"><li>• Account opened.</li></ul>         |

## Opening Test Cases:

\*This table assumes that each case is tried for each one of the four account types.

| <u>Test Case #</u> | <u>Requirement</u>   | <u>Test description and Input Data</u>  | <u>Expected Result/Output</u>  |
|--------------------|--|---|--|
| 1                  | Account should not be opened if create account is pressed with no data | <ul style="list-style-type: none"> <li>Create an instance of Account with no data</li> <li>test data: "O C"</li> </ul>  | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>     |
| 2                  | Account should not be opened if only first or last name is entered     | <ul style="list-style-type: none"> <li>Create an instance of Account with only first or last name entered</li> <li>test data: "O C John"</li> <li>"O C Doe"</li> </ul>  | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>     |
| 3                  | Account should not be opened if just the full name is put in           | <ul style="list-style-type: none"> <li>Create an instance of Account with full name put in</li> <li>test data: "O C John Doe"</li> </ul>  | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>     |
| 4                  | Account should not be opened if the full name and DOB is put in        | <ul style="list-style-type: none"> <li>Create an instance of Account with full name and DOB entered</li> <li>test data: "O C John Doe 2/19/1989"</li> </ul>   | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>     |
| 5                  | Account should not be opened if invalid deposit (null, zero, negative) | <ul style="list-style-type: none"> <li>Create three instances of Account with invalid deposit</li> <li>test data: "O CC John Doe 2/19/1989 X 0"</li> <li>"O CC John Doe 2/19/1989 -100 0"</li> <li>"O CC John Doe 2/19/1989 0 0"</li> </ul> | false <ul style="list-style-type: none"> <li>Initial deposit cannot be 0 or negative.</li> </ul> |
| 6                  | Account should not be opened if missing campus code                    | <ul style="list-style-type: none"> <li>Create an instance of Account with valid data but missing campus code</li> <li>test data: "O CC John Doe 2/19/1989 0"</li> </ul>   | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>     |
| 7                  | Account is successfully opened   | <ul style="list-style-type: none"> <li>Create an instance of Account with</li> </ul>  | true <ul style="list-style-type: none"> <li>Account opened.</li> </ul>                           |

|    |  |  |   |
|----|--|--|---|
|    |  | <ul style="list-style-type: none"> <li>valid data</li> <li>test data: "O CC John Doe 2/19/1989 599.99 0"</li> </ul>  |   |
| 8  | Account is successfully reopened                             | <ul style="list-style-type: none"> <li>After creating an account and it has been closed, it is still in the database.</li> <li>test data: "O CC John Doe 2/19/1989 599.99 0"</li> </ul>  | true <ul style="list-style-type: none"> <li>Account reopened.</li> </ul>  |
| 9  | Account should not open if duplicate is in database already  | <ul style="list-style-type: none"> <li>Create an instance of Account that is a duplicate of previously created account</li> <li>test data: "O CC John Doe 2/19/1989 599.99 0"</li> </ul> | false <ul style="list-style-type: none"> <li>John Doe 2/19/1989 same account(type) is in the database.</li> </ul> |
| 10 | Account should not be opened if no account type is selected. | <ul style="list-style-type: none"> <li>Create an instance of Account that has to account type selected.</li> <li>test data: "O John Doe 2/19/1989 599.99 0"</li> </ul>                   | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>                      |
| 11 | Account should not be opened if just DOB is entered.         | <ul style="list-style-type: none"> <li>Create an instance of Account that has just DOB entered.</li> <li>test data: "O CC 2/19/1989 599.99 0"</li> </ul>                                 | false <ul style="list-style-type: none"> <li>Missing data for opening an account.</li> </ul>                      |

## Closing Test Cases:

\*This table assumes that each case is tried for each one of the four account types.

| <u>Test Case #</u> | <u>Requirement</u>  | <u>Test description and Input Data</u>   | <u>Expected Result/Output</u>  |
|--------------------|---|--|--|
| 1                  | Account should not be closed if close account is pressed with no data | <ul style="list-style-type: none"><li>• Create an instance of Account with no data</li><li>• test data: "C C"</li></ul>  | false <ul style="list-style-type: none"><li>• Missing data for closing an account.</li></ul> |
| 2                  | Account should not be closed if only first or last name is entered    | <ul style="list-style-type: none"><li>• Create an instance of Account with only first or last name entered</li><li>• test data: "C C John"</li><li>• "C C Doe"</li></ul> | false <ul style="list-style-type: none"><li>• Missing data for closing an account.</li></ul> |
| 3                  | Account should not be closed if just the full name is put in          | <ul style="list-style-type: none"><li>• Create an instance of Account with full name put in</li><li>• test data: "C C John Doe"</li></ul>                                | false <ul style="list-style-type: none"><li>• Missing data for closing an account.</li></ul> |
| 4                  | Account should not be closed if just DOB is put in                    | <ul style="list-style-type: none"><li>• Create an instance of Account with full name and DOB entered</li><li>• test data: "C C 2/19/1989"</li></ul>                      | false <ul style="list-style-type: none"><li>• Missing data for closing an account.</li></ul> |
| 5                  | Account should not be closed if no account type is specified          | <ul style="list-style-type: none"><li>• Create an instance of Account with no account type specified</li><li>• test data: "C John Doe 2/19/1989 1500"</li></ul>          | false <ul style="list-style-type: none"><li>• Missing data for closing an account.</li></ul> |
| 6                  | Account is successfully closed  | <ul style="list-style-type: none"><li>• Create an instance of Account with valid data</li><li>• test data: "C CC Kate Lindsey 8/31/2001"</li></ul>                       | true <ul style="list-style-type: none"><li>• Account closed.</li></ul>                       |
| 7                  | Account is already closed   | <ul style="list-style-type: none"><li>• An account in the database is already closed.</li><li>• test data: "C CC Kate Lindsey 8/31/2001"</li></ul>                       | false <ul style="list-style-type: none"><li>• Account is closed already.</li></ul>           |

## Deposit Test Cases:

\*This table assumes that each case is tried for each one of the four account types.

| <u>Test Case #</u> | <u>Requirement</u>   | <u>Test description and Input Data</u>  | <u>Expected Result/Output</u>  |
|--------------------|--|---|--|
| 1                  | Amount should not be deposited due to invalid amount. (null)     | <ul style="list-style-type: none"><li>• Attempt to deposit an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 AAA"</li><li>• "D S April March 1/15/1987"</li></ul> | false <ul style="list-style-type: none"><li>• Not a valid amount.</li></ul>                                    |
| 2                  | Amount should not be deposited due to invalid amount. (negative) | <ul style="list-style-type: none"><li>• Attempt to deposit an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 -100"</li></ul>                                      | false <ul style="list-style-type: none"><li>• Deposit - amount cannot be 0 or negative.</li></ul>              |
| 3                  | Amount should not be deposited due to invalid amount. (zero)     | <ul style="list-style-type: none"><li>• Attempt to deposit an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 0"</li></ul>   | false <ul style="list-style-type: none"><li>• Deposit - amount cannot be 0 or negative.</li></ul>              |
| 4                  | Successful deposit   | <ul style="list-style-type: none"><li>• Attempt to deposit an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 100"</li></ul>                                       | true <ul style="list-style-type: none"><li>• Deposit - balance updated.</li></ul>                              |
| 5                  | Attempt to deposit amount into an account that does not exist.   | <ul style="list-style-type: none"><li>• Attempt to deposit an amount into an account that is not currently created in the database.</li><li>• test data: "D S Kiernan King 8/18/2001 500"</li></ul>                               | false <ul style="list-style-type: none"><li>• Kiernan King 8/18/2001 Savings is not in the database.</li></ul> |

## Withdraw Test Cases:

\*This table assumes that each case is tried for each one of the four account types.

| <u>Test Case #</u> | <u>Requirement</u>   | <u>Test description and Input Data</u>   | <u>Expected Result/Output</u>  |
|--------------------|--|--|--|
| 1                  | Amount should not be withdrawn due to invalid amount. (null)     | <ul style="list-style-type: none"><li>• Attempt to withdraw an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 AAA"</li><li>• "D S April March 1/15/1987"</li></ul> | false <ul style="list-style-type: none"><li>• Not a valid amount.</li></ul>                                      |
| 2                  | Amount should not be withdrawn due to invalid amount. (negative) | <ul style="list-style-type: none"><li>• Attempt to withdraw an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 -100"</li></ul>                                      | false <ul style="list-style-type: none"><li>• Withdraw - amount cannot be 0 or negative.</li></ul>               |
| 3                  | Amount should not be withdrawn due to invalid amount. (zero)     | <ul style="list-style-type: none"><li>• Attempt to withdraw an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 0"</li></ul>   | false <ul style="list-style-type: none"><li>• Withdraw - amount cannot be 0 or negative.</li></ul>               |
| 4                  | Successful withdrawal  | <ul style="list-style-type: none"><li>• Attempt to withdraw an amount into an account created that is opened in the database.</li><li>• test data: "D S April March 1/15/1987 100"</li></ul>                                       | true <ul style="list-style-type: none"><li>• Withdraw - balance updated.</li></ul>                               |
| 5                  | Attempt to withdraw amount from an account that does not exist.  | <ul style="list-style-type: none"><li>• Attempt to withdraw an amount into an account that is not currently created in the database.</li><li>• test data: "D S Ahmed Alghazwi"</li></ul>   | false <ul style="list-style-type: none"><li>• Ahmed Alghazwi 4/12/2000 Savings is not in the database.</li></ul> |

|   |   |   |   |
|---|---|---|---|
|   |   | 4/12/2000 500"  |   |
| 6 | Attempt to withdraw an amount greater than account balance. | <ul style="list-style-type: none"><li>• Attempt to withdraw an amount greater than balance.</li><li>• test data: "W MM Roy Brooks 10/31/1979 100"</li></ul> | false <ul style="list-style-type: none"><li>• Withdraw - insufficient fund.</li></ul> |

### printBy() methods Test Cases:

| <u>Test Case #</u> | <u>Requirement</u>  | <u>Test description and Input Data</u>   | <u>Expected Result/Output</u>  |
|--------------------|---|--|--|
| 1                  | Empty database is printed.  | <ul style="list-style-type: none"><li>• Navigate to Database tab and press any button with no accounts created.</li><li>• test data: "P"</li></ul>                         | false <ul style="list-style-type: none"><li>• Account Database is empty!</li></ul>   |
| 2                  | List of accounts is printed in current order.                       | <ul style="list-style-type: none"><li>• Navigate to Database tab and press 'Print All Account' button with accounts created.</li><li>• test data: "P"</li></ul>            | true <ul style="list-style-type: none"><li>• *list of accounts in the database*</li><li>• Savings::John Doe 2/19/1990::Balance \$200.00</li><li>• *end of list*</li></ul>  |
| 3                  | List of accounts is printed in order by account type.               | <ul style="list-style-type: none"><li>• Navigate to Database tab and press 'Print All Accounts By Type' button with accounts created.</li><li>• test data: "PT"</li></ul>  | true <ul style="list-style-type: none"><li>• *list of accounts by account type.</li><li>• Savings::John Doe 2/19/1990::Balance \$200.00</li><li>• *end of list.</li></ul>  |
| 4                  | List of accounts is printed in order by updated balance.            | <ul style="list-style-type: none"><li>• Navigate to Database tab and press 'Update Balance' button with accounts created.</li><li>• test data: "UB"</li></ul>              | true <ul style="list-style-type: none"><li>• *list of accounts with updated balance</li><li>• Checking::April March 1/15/1987::Balance \$925.08</li><li>• *end of list.</li></ul>  |
| 5                  | List of accounts is printed in order by fees and monthly interests. | <ul style="list-style-type: none"><li>• Navigate to Database tab and press 'Calculate Interest and Fees' button with accounts created.</li><li>• test data: "PI"</li></ul> | true <ul style="list-style-type: none"><li>• *list of accounts with fee and monthly interest</li><li>• Checking::April March 1/15/1987::Balance \$950.00::fee \$25.00::monthly interest \$0.08</li><li>• *end of list.</li></ul> |



| <b>BankTellerController /Interest and fees:</b> |   |  |                               |
|---|---|--|-------------------------------|
| <b>Test Case #</b>                              | <b>Requirement</b>  | <b>Test description and Input Data</b>   | <b>Expected result/output</b> |
| 1   | The method shall apply interest of 0.1 % on regular checking account                | <ul style="list-style-type: none"> <li>• Instance of regular checking account</li> <li>• test data: Regular Checking Account</li> </ul>                    | 0.1%                          |
| 2   | The method shall apply interest of 0.25 % on college checking account               | <ul style="list-style-type: none"> <li>• Instance of college checking account</li> <li>• test data: College Checking Account</li> </ul>                    | 0.25%                         |
| 3   | The method shall apply interest of 0.45 % on saving account with loyal customer     | <ul style="list-style-type: none"> <li>• Instance of college checking account</li> <li>• test data: College Checking Account and loyal customer</li> </ul> | 0.45%                         |
| 4   | The method shall apply interest of 0.3 % on saving account for non-loyal customer   | <ul style="list-style-type: none"> <li>• Instance of saving account</li> <li>• test data: Saving Account</li> </ul>  | 0.3%                          |
| 5   | The method shall apply interest of 0.8 % on Money Market account                    | <ul style="list-style-type: none"> <li>• Instance of money market account</li> <li>• test data: Money Market Account</li> </ul>                            | 0.8%                          |
| 6   | The method shall apply interest of 0.95% on Money Market account for loyal customer | <ul style="list-style-type: none"> <li>• Instance of money market account</li> <li>• test data: Money Market Account with loyal customer</li> </ul>        | 0.95%                         |

| Account Database/close: |  |  |                        |
|-------------------------|--|--|------------------------|
| Test Case #             | Requirement  | Test description and Input Data  | Expected result/output |
| 1                       | The method shall return false if account does not exist in database and is not closed                  | <ul style="list-style-type: none"> <li>An instance of account which is not present in database.</li> <li>test data: non existing account</li> </ul>                    | false                  |
| 2                       | The method shall return false if account exist in database but is already closed                       | <ul style="list-style-type: none"> <li>An instance of account which is present in database and is closed.</li> <li>test data: existing closed account</li> </ul>       | false                  |
| 3                       | The method shall return true if account exist in database but is not already closed and shall close it | <ul style="list-style-type: none"> <li>An instance of account which is present in database and is not closed.</li> <li>test data: existing unclosed account</li> </ul> | true                   |

| Account Database/open: |  |   |                        |
|------------------------|--|---|------------------------|
| Test Case #            | Requirement  | Test description and Input Data   | Expected result/output |
| 1                      | The method opens a new account and store in database and shall return true | <ul style="list-style-type: none"> <li>Create an instance of account and open account.</li> <li>test data: new account</li> </ul> | true                   |

| Account Database/ checkOpen: |   |   |                        |
|------------------------------|---|---|------------------------|
| Test Case #                  | Requirement   | Test description and Input Data   | Expected result/output |
| 1                            | The method shall return 2 if account already opened   | <ul style="list-style-type: none"> <li>An instance of account which is already exist in database and is open</li> <li>test data: existing opened account</li> </ul>   | 2                      |
| 2                            | The method shall return 1 if account already exists and is closed. It shall reopen the account. | <ul style="list-style-type: none"> <li>An instance of account which is already exist in database and is closed</li> <li>test data: existing closed account</li> </ul> | 1                      |
| 3                            | The method shall return 1 if account does not already exist. It shall open the account.         | <ul style="list-style-type: none"> <li>An instance of account which is not already exist in database</li> <li>test data: non existing account</li> </ul>              | 0                      |

| Account Database/find: |   |  |                        |
|------------------------|---|--|------------------------|
| Test Case #            | Requirement   | Test description and Input Data  | Expected result/output |
| 1                      | The method shall return -1 if account does not exist in database      | <ul style="list-style-type: none"> <li>• Create an instance of account which is not present in database.</li> <li>• test data: non existing account</li> </ul> | -1                     |
| 2                      | The method shall return index of account if account exist in database | <ul style="list-style-type: none"> <li>• Create an instance of account which is present in database.</li> <li>• test data: existing account</li> </ul>         | index                  |