Java Class Name: `Controller` Method Signature: `private void openAccount()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	"SAVINGS account 300027410 has been opened." Note: The last four digits can vary.
2	Open a new College Checking account with the following information: First Name: Bob Last Name: Smith DOB: 1/1/2005 Account Type: College Checking Campus: New Brunswick Branch: Piscataway Deposit Amount: 2000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2005 Account Type: College Checking Campus: New Brunswick Branch: Piscataway Deposit Amount: 2000	COLLEGE_CHECKING account 400040432 has been opened. Note: The last four digits can vary.
3	Open a new Savings account with the following information: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount:	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • There is no balance provided which will result in an error	"For input string "" - not a valid amount."

		• test input: First Name: Bob Last Name: Smith DOB: 1/1/2005 Account Type: College Checking Campus: New Brunswick Branch: Piscataway Deposit Amount:	
4	Open a new Savings account with the following information: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: Twenty Thousand	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • The balance is provided as a String representation that cannot be converted to a number • test input: First Name: Bob Last Name: Smith DOB: 1/1/2005 Account Type: College Checking Campus: New Brunswick Branch: Piscataway Deposit Amount: Twenty Thousand	"For input string "Twenty Thousand" - not a valid amount."
5	Open a new Savings account with the following information: First Name: Bob Last Name: Smith DOB: Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • The date of birth is not provided which will result in an error • test input: First Name: Bob Last Name: Smith DOB: Account Type: Savings	"Date input Invalid"

		Branch: Princeton Deposit Amount: 10000	
6	Open a new Savings account with the following information: First Name: Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● The first name is not provided which will result in an error ● test input: First Name: Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	"Name invalid"
7	Open a new Savings account with the following information: First Name: Alice Last Name: DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● The last name is not provided which will result in an error ● test input: First Name: Alice Last Name: DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	"Name invalid"
8	Open a new Savings account with the following information: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch:	● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● The Branch is not	"Select Branch"

	Deposit Amount: 10000	provided which will result in an error • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Deposit Amount: 10000	
9	Open a new Checking account with the following information: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000 Then attempt to open another Checking account with the following information: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • Attempts to open a duplicate account that is not a Certificate Deposit Account • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000 • After the account has been successfully opened, the user inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	"CHECKING account 300017410 has been opened." Note: The last four digits can vary. After attempting to open another Checking account for Bob Smith at the same branch. "Bob Smith already has a CHECKING account."

Java Class Name: `Controller` **Method Signature:** `private boolean checkDateOfBirth(AccountType accountType, Date dob)`			
Java Class Name: `Date` **Method Signature:** `public boolean isAfterToday()`			
Java Class Name: `Date` **Method Signature:** `public boolean isEighteen()`			
Java Class Name: `Date` **Method Signature:** `public boolean isOverTwentyFour()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information: First Name: Alice Last Name: Smith DOB: 3/30/2025 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • Calls the *isAfterToday* method from the Date class in the *checkDateOfBirth* method • test input: First Name: Alice Last Name: Smith DOB: 3/30/2025 Account Type: Savings Branch: Princeton Deposit Amount: 10000	"DOB invalid: 3/30/2025 cannot be today or a future day."
2	Open a new Savings account with the following information: First Name: Alice Last Name: Smith DOB: 1/1/2021 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • Calls the *isEighteen* method from the Date class in the *checkDateOfBirth* method • test input: First Name: Alice Last Name: Smith DOB: 1/1/2021	"Not eligible to open: 1/1/2021 under 18."

		Account Type: Savings Branch: Princeton Deposit Amount: 10000	
3	Open a new College Checking account with the following information: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: College Checking Campus: New Brunswick Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • Calls the *isOverTwentyFour* method from the Date class in the *checkDateOfBirth* method • test input: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: College Checking Campus: New Brunswick Branch: Princeton Deposit Amount: 10000	"Not eligible to open: 1/1/2000 over 24."

Java Class Name: `Controller` **Method Signature:** `clearArgumentsOpen()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Clear" button: • test input: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	All arguments that the user has inputted in the GUI has been cleared. Account Type also resets to Checking.

Java Class Name: `Controller` Method Signature: `depositMoney()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information and deposit $1000 into that account: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • Navigate to the "Deposit/Withdraw/Close" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Deposit" button: • test input: Account Number: 300017410 Amount: $1000	"CHECKING account 300017410 has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs… "$1,000.00 deposited to 300017410"
2	Open a new Savings account with the following information and deposit $1000 from that account: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000	"CHECKING account 300017410 has been opened." Note: The last four digits can vary.

		Account Type: Savings Branch: Princeton Deposit Amount: 10000 • Navigate to the "Deposit/Withdraw/Close" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Deposit" button: • test input: Account Number: 300017410 Amount: One Thousand	After navigating to second tab and adding inputs… "For input string: "One Thousand" - not a valid amount."

Java Class Name: `Controller` **Method Signature:** `withdrawMoney()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information and withdraw $1000 from that account: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • Navigate to the "Deposit/Withdraw/Close" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Withdraw" button:	"CHECKING account 300017410 has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs… "$1,000.00 withdrawn to 300017410"

		• test input: Account Number: 300017410 Amount: $1000	
2	Open a new Savings account with the following information and withdraw $1000 from that account: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • Navigate to the "Deposit/Withdraw/Close" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Withdraw" button: • test input: Account Number: 300017410 Amount: One Thousand	"CHECKING account 300017410 has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs… "For input string: "One Thousand" - not a valid amount."

Java Class Name: `Controller` **Method Signature:** `private void closeSingleAccount()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information and then close the account: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob	"CHECKING account 300016899 has been opened." Note: The last four digits

	Deposit Amount: 10000	Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • Navigate to the "Deposit/Withdraw/Close" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Close" button: • test input: Account Number: 300016899 Date Close: 3/21/2025	can vary. After navigating to second tab and adding inputs… "Closing account 300016899 --interest earned: $8.63"

Java Class Name: `Controller` **Method Signature:** `private void closeAllAccounts()`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Open a new Savings account with the following information and then a Checking account with the following information then close the accounts: Savings First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 Checking First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input:	"SAVINGS account 30002**6899** has been opened. CHECKING account 30001**7410** has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs…

		First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000 ● Navigate to the "Deposit/Withdraw/Close" tab ● User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Close All" button: ● test input: Date Close: 3/21/2025 First Name: Bob Last Name: Smith DOB: 1/1/2000	"Closing accounts for Bob Smith 1/1/2000 --300026899 interest earned: $15.82 --300017410 interest earned: $8.63 All accounts for Bob Smith 1/1/2000 are closed and moved to archive."
2	Open a new Savings account with the following information and then a Checking account with the following information then close the accounts: Savings First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 Checking First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 ● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking	"SAVINGS account 300026899 has been opened. CHECKING account 300017410 has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs… "Name Invalid"

		Branch: Princeton Deposit Amount: 10000 ● Navigate to the "Deposit/Withdraw/Close" tab ● User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Close All" button: ● The argument is missing the first name ● test input: Date Close: 3/21/2025 First Name: Last Name: Smith DOB: 1/1/2000	
3	Open a new Savings account with the following information and then a Checking account with the following information then close the accounts: Savings First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 Checking First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 ● User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	"SAVINGS account 300026899 has been opened. CHECKING account 300017410 has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs… "Name Invalid"

		• Navigate to the "Deposit/Withdraw/Close" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Close All" button: • The argument is missing the last name • test input: Date Close: 3/21/2025 First Name: Bob Last Name: DOB: 1/1/2000	
4	Open a new Savings account with the following information and then a Checking account with the following information then close the accounts: Savings First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 Checking First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Checking Branch: Princeton Deposit Amount: 10000 • Navigate to the "Deposit/Withdraw/Cl	"SAVINGS account 300026899 has been opened. CHECKING account 300017410 has been opened." Note: The last four digits can vary. After navigating to second tab and adding inputs… "Date input Invalid"

		ose" tab • User inputs the following fields into the GUI in the "Deposit/Withdraw/Close" tab before clicking the "Close All" button: • The argument is missing the last name • test input: Date Close: First Name: Bob Last Name: DOB: 1/1/2000	

Java Class Name: `Controller` **Method Signature:** `private void printByHolder()`			
Java Class Name: `Sort` **Method Signature:** `public static void account(AccountDatabase list, char key)`			
Test Case #	**Functional Requirement, or Test Objective**	**Test Description and Input Data**	**Expected result/output**
1	Printing all the Accounts in the Account Database when there are no Accounts that have been added	Navigate to the tab labeled "Account Management" and click the "Print by Holder" button. No Accounts should be printed.	*List of accounts ordered by account holder and number. *end of list.
2	Open a new account for one holder with the following information and then another account for a different holder with the following information, then print by holder: Bob First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 Alice	• User inputs the following fields into the GUI in the "Open Account" tab before clicking the "Open" button: • test input: First Name: Bob Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 • User inputs the	"SAVINGS account 300027410 has been opened. SAVINGS account 300026899 has been opened." Note: The last four digits can vary.

	First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000	following fields into the GUI in the "Open Account" tab before clicking the "Open" button: ● test input: First Name: Alice Last Name: Smith DOB: 1/1/2000 Account Type: Savings Branch: Princeton Deposit Amount: 10000 ● Navigate to the "Account Management" tab ● User presses the Print by Holder button The Accounts printed should be ordered by holder	After navigating to third tab and clicking button… "*List of accounts ordered by account holder and number. Account#[300026899] Holder[Alice Smith 1/1/2000] Balance[$10,000.00] Branch[PRINCETON] Account#[300027410] Holder[Bob Smith 1/1/2000] Balance[$10,000.00] Branch[PRINCETON] *end of list."