**LOGIN**

**In a CLASS called MainActivity:**

CREATE an INTEGER variable called loginCounter, and 2 STRING variables called username and password.

In the onCreate method

CREATE 2 FINAL STRINGS called staticUsername and staticPassword to hold onto hardcoded username and password. ASSIGN staticUsername the value of ‘a’ and ASSIGN staticPassword the value of ‘b’.

ASSIGN loginCounter the value of 3, to represent the 3 max login attempts.

CREATE a FINAL EditText reference variable called usernameET and assign it the usernameET edittext field.

CREATE a FINAL EditText reference variable called passwordET and assign it the passwordET edittext field.

CREATE a FINAL Button reference variable called loginB and assign it the loginButton button field from MainActivity Layout.

CREATE an onClickListener function for loginB

GRAB and ASSIGN username and password from usernameET and passwordET

IF loginCounter is greater than 1

CHECK to make sure username and password are not empty

IF empty

Prompt user to fill in fields to continue

ELSEIF not empty CHECK to see if username and password match whats on file

IF username and password match staticPassword and staticUsername

SEND user to userHome CLASS via intents

ELSEIF username and password do not match

DECREMENT loginCounter

ALERT user that credentials are wrong and tell them how many attempts they have left

ENDIF

ENDIF

ELSEIF loginCounter is not greater than 1

START a 10 second countdown timer and ALERT user that max attempts are exceeded and the app is shutting down

ENDIF

CREATE a function called onCreateOptionsMenu

CALL the getMenuInflater function and inflate the menu activity to action bar

CREATE a function called onOptionsItemSelected to handle actionbar item

CREATE an integer called id to hold the menu item id

CREATE a SWITCH statement

IF QUIT option is selected

KILL all app processes and shut down

ENDIF

**USERHOME**

**In a CLASS called userHome:**

In the onCreate method:

CALL checkingButton() and savingsButton() functions

CREATE a function called checkingButton

CREATE a FINAL Button reference variable called checkingB and assign it the checkingButton button field from userHome Layout.

CREATE an onClickListener function for checkingB

SEND user to CheckingActivity CLASS via intents

CREATE a function called checkingButton

CREATE a FINAL Button reference variable called savingsB and assign it the savingsButton button field from userHome Layout.

CREATE an onClickListener function for savingsB

SEND user to SavingsActivity CLASS via intents

CREATE a function called onCreateOptionsMenu

CALL the getMenuInflater function and inflate the menu activity to action bar

CREATE a function called onOptionsItemSelected to handle actionbar item

CREATE an integer called id to hold the menu item id

CREATE a SWITCH statement

IF QUIT option is selected

KILL all app processes and shut down

ENDIF

**CHECKING ACTIVITY**

**In a CLASS called CheckingActivity:**

CREATE a PUBLIC FINAL STATIC STRING called MY\_PREFS and assign it the value “my\_prefs”

CREATE a PUBLIC FINAL STATIC STRING called CHECKING\_KEY and assign it the value “checking\_key";

CREATE a PUBLIC FINAL STATIC STRING called SAVINGS\_KEY and assign it the value “savings\_key";

CREATE a SHAREPREFERENCES EDITOR called myEditor

CREATE PUBLIC doubles called checkingBalance, savingsBalance, and tempBalance

CREATE PUBLIC ints emptyField, radioButtonSelected

In the onCreate method:

Initialize radioButtonselected to 0

CALL updateCheckingBalanceTV() and submitButton() functions

CREATE a function called updateCheckingBalanceTV()

CREATE TEXTVIEW variable called checkBalanceTV and assign to the checkingBalanceTV textview in the CheckingActivity layout.

CREATE SHAREPREFERENCES variable called checkingPref and get the SharePreferences and assign to checkingPref

GET CHECKING\_KEY value from sharedPreferences and convert the value to a double and assign to checkingBalance variable

CREATE a decimal format object to format the checkingBalance value to a monetary value

CREATE a STRING called checkingBalanceString and assign it the formatted checkingBalance

SET checkingBalanceTV textView to the checkingBalanceString

CREATE a function called withdrawChecking()

CREATE EDITTEXT variable called userInput and assign it to the checkingEditText EditText in the CheckingActivity layout.

CREATE a double variable called userInputValue and assign it the the text from userInput and convert to a double

IF the currentbalance is greater than the value in userInputValue

DEDUCT userInputValue from currentbalance

ELSE Alert user of insufficient funds

RETURN the currentbalance

ENDIF

CREATE a function called depositChecking()

CREATE EDITTEXT variable called userInput and assign it to the checkingEditText EditText in the CheckingActivity layout.

CREATE a double variable called userInputValue and assign it the the text from userInput and convert to a double

ADD userInputValue to currentbalance

RETURN the currentbalance

CREATE a function called updateChecking()

GET sharedPreferences and place into myEditor

CONVERT tempBalance to a long and place into CHECKING\_KEY

checkingBalance equals tempBalance value

CREATE a function called submitButton()

CREATE FINAL button variable called submitB and assign it to the checkingSubmitbutton BUTTON in the CheckingActivity layout.

CREATE an onClickListener function for submitB

Assign emptyField variable the return value from checkNotEmpty() function

IF emptyField equals 0

Alert user to enter a money amount in textbox

ELSE

IF radioButtonSelected equals 0

Alert user to make a radio button choice

ELSEIF radioButtonSelected equals 1

CALL withdrawChecking() function and assign return value to tempBalance

CALL updateChecking() and updateCheckingBalanceTV() function

ELSEIF radioButtonSelected equals 2

CALL depositChecking() function and assign return value to tempBalance

CALL updateChecking() and updateCheckingBalanceTV() function

ELSEIF radioButtonSelected equals 3

CALL withdrawChecking() function and assign return value to tempBalance

GET sharePreferences and place in savingsPref

GET value from SAVINGS\_KEY and convert to a double and place in savingsBalance

CREATE EDITTEXT variable called userInput and assign it to the checkingEditText EditText in the CheckingActivity layout.

CREATE a double variable called userInputValue and assign it the the text from userInput and convert to a double

ADD userInputValue to savingsBalance

CONVERT savingsBalance to long and place into SAVINGS\_KEY in sharedPreferences

CALL updateChecking() and updateCheckingBalanceTV() function

ENDIF

ENDIF

CREATE a function called onRadioButtonClicked

CREATE a Boolean variable called checked and assign it the value of the radiobutton.isChecked

CREATE a switch statement for radiobuttons

CASE withdrawCRB radiobutton is checked

Assign radioButtonSelected 1

CASE depositCRB radiobutton is checked

Assign radioButtonSelected 2

CASE transferCRB radiobutton is checked

Assign radioButtonSelected 3

CREATE a function called checkNotEmpty

CREATE EDITTEXT variable called userInput and assign it to the checkingEditText EditText in the CheckingActivity layout.

CREATE a STRING called struserInput and assign it the text from userInput

CREATE an int called empty

IF struserIput is empty

SET empty to 0

ELSE

SET empty to 1

RETURN empty

ENDIF

CREATE a function called onCreateOptionsMenu

CALL the getMenuInflater function and inflate the menu activity to action bar

CREATE a function called onOptionsItemSelected to handle actionbar item

CREATE an integer called id to hold the menu item id

CREATE a SWITCH statement

IF QUIT option is selected

KILL all app processes and shut down

ENDIF

**SAVINGS ACTIVITY**

**In a CLASS called SavingsActivity:**

CREATE a PUBLIC FINAL STATIC STRING called MY\_PREFS and assign it the value “my\_prefs”

CREATE a PUBLIC FINAL STATIC STRING called CHECKING\_KEY and assign it the value “checking\_key";

CREATE a PUBLIC FINAL STATIC STRING called SAVINGS\_KEY and assign it the value “savings\_key";

CREATE a SHAREPREFERENCES EDITOR called myEditor

CREATE PUBLIC doubles called checkingBalance, savingsBalance, and tempBalance

CREATE PUBLIC ints emptyField, radioButtonSelected

In the onCreate method:

Initialize radioButtonselected to 0

CALL updateSavingsBalanceTV() and submitButton() functions

CREATE a function called updateSavingsBalanceTV()

CREATE TEXTVIEW variable called saveBalanceTV and assign to the savingsBalanceTV textview in the SavingsActivity layout.

CREATE SHAREPREFERENCES variable called savingsPref and get the SharePreferences and assign to savingsPref

GET SAVINGS\_KEY value from sharedPreferences and convert the value to a double and assign to savingsBalance variable

CREATE a decimal format object to format the savingsBalance value to a monetary value

CREATE a STRING called savingsBalanceString and assign it the formatted savingsBalance

SET savingsBalanceTV textView to the savingsBalanceString

CREATE a function called withdrawSavings ()

CREATE EDITTEXT variable called userInput and assign it to the savingsEditText EditText in the SavingsActivity layout.

CREATE a double variable called userInputValue and assign it the the text from userInput and convert to a double

IF the currentbalance is greater than the value in userInputValue

DEDUCT userInputValue from currentbalance

ELSE Alert user of insufficient funds

RETURN the currentbalance

ENDIF

CREATE a function called depositSavings ()

CREATE EDITTEXT variable called userInput and assign it to the savingsEditText EditText in the SavingsActivity layout.

CREATE a double variable called userInputValue and assign it the the text from userInput and convert to a double

ADD userInputValue to currentbalance

RETURN the currentbalance

CREATE a function called updateSavings ()

GET sharedPreferences and place into myEditor

CONVERT tempBalance to a long and place into SAVINGS\_KEY

checkingBalance equals tempBalance value

CREATE a function called submitButton()

CREATE FINAL button variable called submitB and assign it to the savingsSubmitbutton BUTTON in the SavingsActivity layout.

CREATE an onClickListener function for submitB

Assign emptyField variable the return value from checkNotEmpty() function

IF emptyField equals 0

Alert user to enter a money amount in textbox

ELSE

IF radioButtonSelected equals 0

Alert user to make a radio button choice

ELSEIF radioButtonSelected equals 1

CALL withdrawSavings () function and assign return value to tempBalance

CALL updateSavings () and updateSavingsBalanceTV() function

ELSEIF radioButtonSelected equals 2

CALL depositSavings () function and assign return value to tempBalance

CALL updateSavings () and updateSavingsBalanceTV() function

ELSEIF radioButtonSelected equals 3

CALL withdrawSavings () function and assign return value to tempBalance

GET sharePreferences and place in checkingPref

GET value from CHECKING\_KEY and convert to a double and place in checkingBalance

CREATE EDITTEXT variable called userInput and assign it to the savingsEditText EditText in the SavingsActivity layout.

CREATE a double variable called userInputValue and assign it the the text from userInput and convert to a double

ADD userInputValue to checkingBalance

CONVERT checkingBalance to long and place into CHECKING\_KEY in sharedPreferences

CALL updateSavings () and updateSavingsBalanceTV() function

ENDIF

ENDIF

CREATE a function called onRadioButtonClicked

CREATE a Boolean variable called checked and assign it the value of the radiobutton.isChecked

CREATE a switch statement for radiobuttons

CASE withdrawSRB radiobutton is checked

Assign radioButtonSelected 1

CASE depositSRB radiobutton is checked

Assign radioButtonSelected 2

CASE transferSRB radiobutton is checked

Assign radioButtonSelected 3

CREATE a function called checkNotEmpty

CREATE EDITTEXT variable called userInput and assign it to the savingsEditText EditText in the SavingsActivity layout.

CREATE a STRING called struserInput and assign it the text from userInput

CREATE an int called empty

IF struserIput is empty

SET empty to 0

ELSE

SET empty to 1

RETURN empty

ENDIF

CREATE a function called onCreateOptionsMenu

CALL the getMenuInflater function and inflate the menu activity to action bar

CREATE a function called onOptionsItemSelected to handle actionbar item

CREATE an integer called id to hold the menu item id

CREATE a SWITCH statement

IF QUIT option is selected

KILL all app processes and shut down

ENDIF