**PROJECT Bank account simulator program**

**Objective** To write a program that performs various bank transactions.

***PROJECT DESCRIPTION***

Bank of IIT has contacted you to write, compile and execute a complete program that creates bank account information and executes various transaction details for their clients.

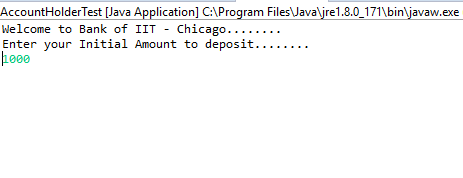
Your program will prompt users for options such as creating an initial balance, entering deposits or withdrawals. Also, your program will allow for the printing of account information including interest at various interest rates.

Use loops, user defined methods, conditional and relational logic and the basics of OOP to accomplish the objectives of this program.

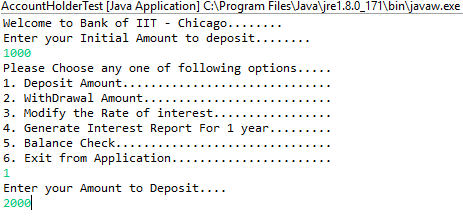
Error trapping will be part of your grade so don’t forget to include some basic error trapping logic! Comment your code thoroughly as well for maximum points.

***Snapshot of Testcases:***

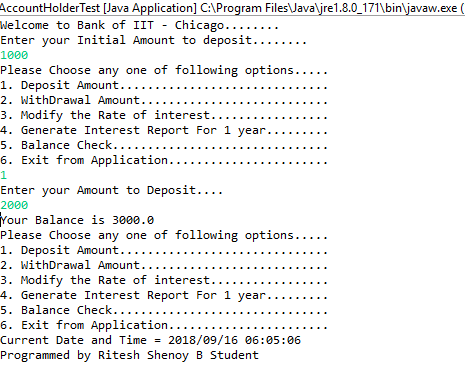
***Prompt for Initial Deposit:***

****

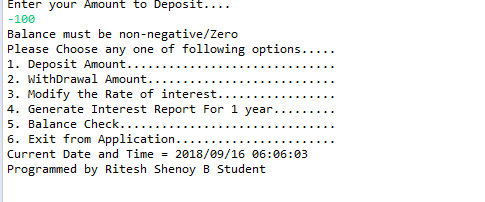
***Prompt for Initial Balance:***



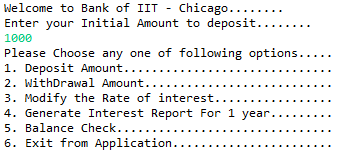
***Shows Deposit of $2000 is successful:***

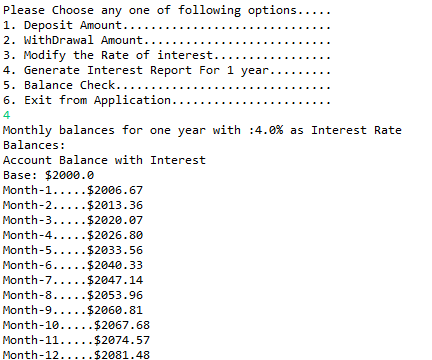
******

***Throw an exception if Initial Deposit is 0 or less than 0:***

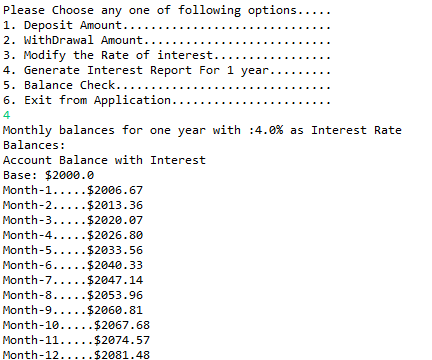


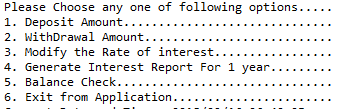
***Prompt the user to Enter his/her choice:***

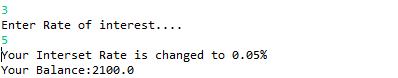
******

***Prompts the user to Enter an option and calculate the ROI for 12 months:***

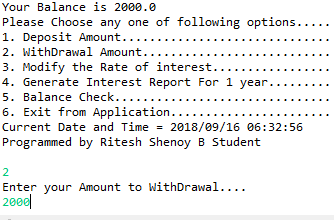
***Provide breakdown of the account balance for 12 months and Update ROI to 5% and display balance:***

******

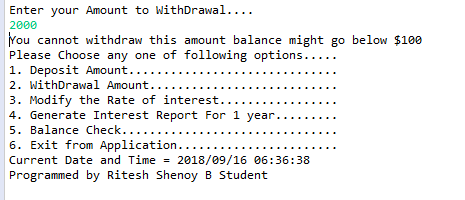
******

******

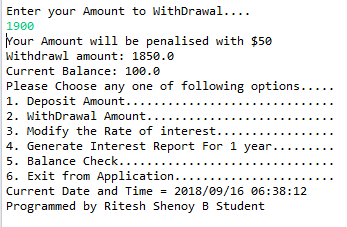
***Prompt the user to Withdraw:***



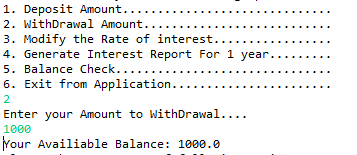
***Don’t allow if the balance goes below $100:***



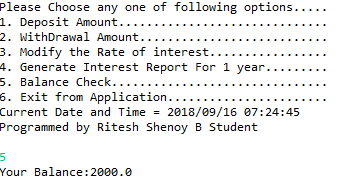
***Reduce $50 if minimum balance of $500 is not met:***



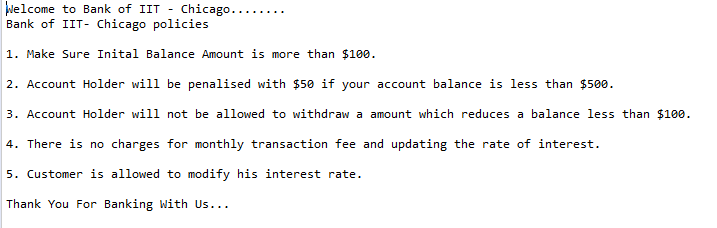
***Prompt for a normal withdraw if the Account balance doesn’t go below $500 or $100:***

******

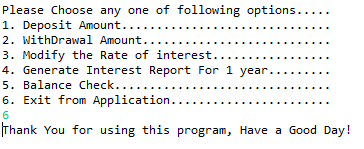
***Display balance if User Enters the option:***

******

***To Display Bank of IIT Policies:***

******

***To Exit the program:***

******

***Source Code for the Lab – I:***

***AccountHolderTest.java***

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Scanner;

/\*AccountHolderTest class

\* created by rshenoy

\* Ritesh Shenoy B

\* Lab 1 ITMD 510\*/

public class AccountHolderTest {

public static void main(String[] args) {

// TODO Auto-generated method stub

//Field create to collect initial amount from the account holder

double initalAmount = 0.0;

//Initially set the interest as 4 %

AccountHolder.annualInterestRate = 0.04;

System.out.println("Welcome to Bank of IIT - Chicago........");

System.out.println("Bank of IIT- Chicago policies\n");

System.out.println("1. Make Sure Inital Balance Amount is more than $100.\n");

System.out.println("2. Account Holder will be penalised with $50 if your account balance is less than $500.\n");

System.out.println("3. Account Holder will not be allowed to withdraw a amount which reduces a balance less than $100.\n");

System.out.println("4. There is no charges for monthly transaction fee and updating the rate of interest.\n");

System.out.println("5. Customer is allowed to modify his interest rate.\n");

System.out.println("Thank You For Banking With Us... \n");

System.out.println("Enter your Initial Amount to deposit........");

Scanner scannerObjectForInputs = new Scanner(System.in);

//Given input for the initial amount

String userInput = scannerObjectForInputs.nextLine();

//Check if the initial amount is null

if(userInput == null) {

System.out.println("Not a Valid Value");

} else {

initalAmount = Double.parseDouble(userInput);

}

//Check if the initial amount is less or equal to 0.0

if(initalAmount <= 0.0) {

System.out.println("Balance must be non-negative/Zero");

}

//Account Holder class object created to acces the class function and fields

AccountHolder accountTestObject = new AccountHolder(initalAmount);

//Function call to display menu option page

accountTestObject.menuOptions();

while(true) {

try {

//Get the input from the user from the menu options asked

int choiceValue = Integer.parseInt(scannerObjectForInputs.nextLine());

switch (choiceValue) {

case 1: //Deposit Amount

//Case for deposit a valid amount

System.out.println("Enter your Amount to Deposit....");

String tempDepositAmount = scannerObjectForInputs.nextLine();

if (tempDepositAmount.equals("")) {

System.out.println("Enter your a valid Deposit amount....");

} else {

double depositAmount = Double.parseDouble(tempDepositAmount);

accountTestObject.depositMoney(depositAmount);

}

System.out.println("Your Balance is " + accountTestObject.mainBalance );

//Function call to display menu option page

accountTestObject.menuOptions();

break;

case 2: //WithDrawal Amount

//Case for withdrawal of a valid amount

System.out.println("Enter your Amount to WithDrawal....");

String tempWithDrawalAmount = scannerObjectForInputs.nextLine();

if (tempWithDrawalAmount.equals("")) {

System.out.println("Enter your a valid withdrawal amount....");

} else {

double withdrawalAmount = Double.parseDouble(tempWithDrawalAmount);

if(withdrawalAmount <= 0.0) {

System.out.println("Balance must be non-negative");

} else {

accountTestObject.withDrawlMoney(withdrawalAmount);

}

}

//Function call to display menu option page

accountTestObject.menuOptions();

break;

case 3: //Modify the Rate of interest

//Case to modify the current rate of interest for the account holder

System.out.println("Enter Rate of interest....");

double rateOfInterest = Double.parseDouble(scannerObjectForInputs.nextLine());

//Call the function modify monthly interest

AccountHolder.modifymonthlyInterest(rateOfInterest/100.0);

System.out.println("Your Balance:" + accountTestObject.mainBalance);

//Function call to display menu option page

accountTestObject.menuOptions();

break;

case 4: //Generate Interest Report For 1 year

//Case to show the amount with interest for every month interval for a year

System.out.println("Monthly balances for one year with :" + (AccountHolder.annualInterestRate \* 100) + "% as Interest Rate");

System.out.println("Balances:");

System.out.println("Account Balance with Interest");

System.out.println("Base: $" + accountTestObject.mainBalance);

double tempAmountForMonth = accountTestObject.mainBalance;

for (int i = 1;i <= 12;i++) {

double tempRateOfInterest = tempAmountForMonth \* (AccountHolder.annualInterestRate /12.0);

tempAmountForMonth = tempAmountForMonth + tempRateOfInterest;

System.out.println("Month-" + i + ".....$" + String.format("%.2f", tempAmountForMonth));

}

//Update the MainBalance with calculated ROI for 12 months value

accountTestObject.mainBalance = Double.parseDouble(String.format("%.2f", tempAmountForMonth));

//Function call to display menu option page

accountTestObject.menuOptions();

break;

case 5: //Balance Check

//Case to display current balance

System.out.println("Your Balance:" + accountTestObject.mainBalance);

//Function call to display menu option page

accountTestObject.menuOptions();

break;

case 6: //Exit from Application

//Good Bye Message

System.out.println("Thank You for using this program, Have a Good Day!");

scannerObjectForInputs.close();

//Case to exit the program

System.exit(0);

break;

default:

System.out.println("Please Press some other Option");

//Function call to display menu option page

accountTestObject.menuOptions();

break;

}

//To display the current timestamp and the Programmer info

String timeStamp = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(Calendar.getInstance().getTime());

System.out.println("Current Date and Time = " + timeStamp + "\nProgrammed by Ritesh Shenoy B\n");

}

catch (Exception e) {

// TODO: handle exception

System.out.println(""+e);

}

}

}

}

**AccountHolder.java**

/\*AccountHolderTest class

\* created by rshenoy

\* Ritesh Shenoy B

\* Lab 1 ITMD 510\*/

public class AccountHolder {

//Field for storing annual interest rate

static double annualInterestRate;

//Field for mainBalnce in user account

double mainBalance = 0.0;

//Parameterized Constructor with initial balance as a argument

public AccountHolder(double mainBalance) {

this.mainBalance = mainBalance;

}

//Deposit money Function

public void depositMoney(double depositMoney) {

if(depositMoney <= 0.0) {

System.out.println("Balance must be non-negative/Zero");

} else {

mainBalance = mainBalance + depositMoney;

}

}

//WithDraw money Function with all the check

public void withDrawlMoney(double withDrawAmount) {

double tempBalance = mainBalance;

mainBalance = mainBalance - withDrawAmount;

//Check the balance is less than 100

if (mainBalance < 100 ) {

System.out.println("You cannot withdraw this amount balance might go below $100");

mainBalance = tempBalance;

} else if(mainBalance > 150 && mainBalance <= 500) { //Check if the balance amount comes between 150 to 500 and if yes penalize with $50

System.out.println("Your Amount will be penalised with $50");

mainBalance = mainBalance - 50; //$50 deduction for withdrawing when having a balance less than 500

System.out.println("Current Balance: " +mainBalance);

} else if(mainBalance >=100 && mainBalance <=150 ) { //Check if the balance amount is less than 150, then penalize them with $50 in the withdrawal amount

System.out.println("Your Amount will be penalised with $50");

if(withDrawAmount <=50) {

System.out.println("You cannot withdraw this amount");

mainBalance = tempBalance;

} else {

withDrawAmount = withDrawAmount - 50;

System.out.println("Withdrawl amount: "+withDrawAmount);

}

System.out.println("Current Balance: " +mainBalance);

} else {

System.out.println("Your Availiable Balance: " +mainBalance);

}

}

//Function to calculate the monthly rate of interest

public void monthlyInterest() {

mainBalance += mainBalance \* (annualInterestRate / 12.0);

}

//Function to change the current rate of interest

public static void modifymonthlyInterest(double newInterestRate) {

if (newInterestRate <= 0.0 && newInterestRate > 10) {

System.out.println("Interest cannot be less than or equal to 0 or greater than 10");

} else {

annualInterestRate = newInterestRate;

System.out.println("Your Interset Rate is changed to " +annualInterestRate + "%");

}

}

//Function call to display menu option page

public void menuOptions() {

System.out.println("Please Choose any one of following options..... ");

System.out.println("1. Deposit Amount.............................. ");

System.out.println("2. WithDrawal Amount........................... ");

System.out.println("3. Modify the Rate of interest................. ");

System.out.println("4. Generate Interest Report For 1 year......... ");

System.out.println("5. Balance Check............................... ");

System.out.println("6. Exit from Application....................... ");

}

}