SHIFA KHAN
21BBS0255

JAVA PROGRAMMING
LA - 4

Q1 *Create a class account that maintains acc_no, name, and balance. Perform deposit, withdrawal and statement print operations. Use subclasses loan account, credit card, and its relevant variables, functions related to it. The class must be able to throw an INSUFFICIENT FUND exception when a user is trying to withdraw a amount less the minimum balance.*

CODE:

```
import java.util.Scanner;


class InsufficientFundsException extends Exception {
   public InsufficientFundsException(String message) {
      super(message);
   }
}


class BankAccount {
   int accountNumber;
   String accountHolderName;
   double balance;
   static int accountNumberCount = 0;

   BankAccount(String accountHolderName, double balance) {
      accountNumber = ++accountNumberCount;
      this.accountHolderName = accountHolderName;
      this.balance = balance;
   }

   BankAccount(String accountHolderName) {
      accountNumber = ++accountNumberCount;
      this.accountHolderName = accountHolderName;
      balance = 0.0;
   }
```

```
    void deposit(double amount) {
        this.balance += amount;
        System.out.println("Amount: $" + amount + " deposited successfully!");
        System.out.println("Your balance is: $" + this.balance);
    }

    void withdraw(double amount) throws InsufficientFundsException {
        if (amount <= this.balance) {
            this.balance -= amount;
            System.out.println("Amount: $" + amount + " withdrawn successfully");
            System.out.println("Your balance is: $" + this.balance);
        } else {
            System.out.println("Insufficient balance! Withdrawal declined!!");
            System.out.println("Your balance is: $" + this.balance);
            throw new InsufficientFundsException("Insufficient balance to withdraw.");
        }
    }

    void printStatement() {
        System.out.println("**************************************************");
        System.out.println("Bank Statement: ");
        System.out.println("Account Number: " + this.accountNumber);
        System.out.println("Account Holder's Name: " + this.accountHolderName);
        System.out.println("Account balance: $" + this.balance);
        System.out.println("--------------------------------------------------");
    }
}

class LoanAccount extends BankAccount {
    float creditScore;
    static double loanAmount = 0;

    LoanAccount(String accountHolderName, double balance, float creditScore) {
        super(accountHolderName, balance);
```

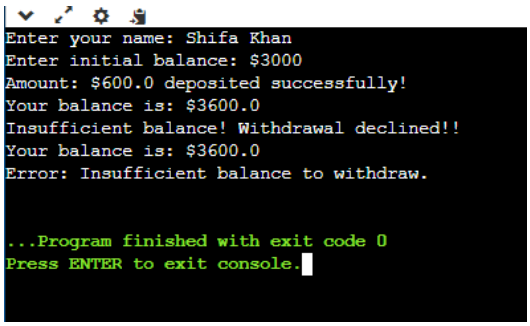
```
    this.creditScore = creditScore;
}

boolean loanApproval(double loanAmount) {
    if (loanAmount > 1000000) {
        if (creditScore > 900) {
            System.out.println("Congratulations, Your loan is approved!!");
            System.out.println("Loan Amount: $" + loanAmount);
            return true;
        } else {
            System.out.println("Not enough credit score.. Loan request declined!");
            return false;
        }
    } else {
        if (creditScore > 700) {
            System.out.println("Congratulations, Your loan is approved!!");
            System.out.println("Loan amount: $" + loanAmount);
            return true;
        }
    }
    System.out.println("Loan not granted.. Credit score is low");
    return false;
}

void getLoan(double loanAmount) {
    loanAmount += loanAmount;
    if (this.loanApproval(loanAmount)) {
        System.out.println("Loan granted worth: $" + loanAmount);
    }
}

int repayLoan(double repayAmount) {
    if (loanAmount == 0) {
        System.out.println("No loans left to repay");
```

```
            return 0;
        }
        if (repayAmount >= loanAmount) {
            loanAmount = 0.0;
            System.out.println("Loan repaid!!");
        } else {
            loanAmount -= repayAmount;
            System.out.println("Repay was successful!!");
            System.out.println("Remaining repay amount: $" + loanAmount);
        }
        return 1;
    }

    void printStatement() {
        super.printStatement();
        System.out.println("Your credit score: " + this.creditScore);
        System.out.println("Your loan amount: $" + this.loanAmount);
        System.out.println("----------------------------------------------------");
    }
}

class CreditAccount extends BankAccount {
    static double creditAmount = 0;
    double maxCreditAmount;

    CreditAccount(String accountHolderName, double balance, double maxCreditAmount) {
        super(accountHolderName, balance);
        this.maxCreditAmount = maxCreditAmount;
    }

    void getCredit(double amount) {
        if (creditAmount + amount <= this.maxCreditAmount) {
            System.out.println("Credit granted worth: $" + amount);
            creditAmount += amount;
```

```
            creditAmount += amount * 0.1;
        } else {
            System.out.println("Credit limit exceeded. Can't retrieve funds.");
        }
    }

    void repayCredit(double repayAmount) {
        creditAmount = creditAmount;
        if (repayAmount >= creditAmount) {
            creditAmount = 0.0;
            System.out.println("Credit repaid!!");
        } else {
            creditAmount -= repayAmount;
            System.out.println("Repay was successful!!");
            System.out.println("Remaining repay amount: $" + creditAmount);
        }
    }

    void printStatement() {
        super.printStatement();
        System.out.println("Your credit amount: $" + this.creditAmount);
        System.out.println("-------------------------------------------------");
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter your name: ");
            String name = scanner.nextLine();

            System.out.print("Enter initial balance: $");
```

```
            double initialBalance = scanner.nextDouble();

            BankAccount userAccount = new BankAccount(name, initialBalance);
            userAccount.deposit(600.0);
            userAccount.withdraw(6000.0);
            userAccount.printStatement();

            System.out.println();

            LoanAccount userLoanAccount = new LoanAccount(name, 5000.0, 800);
            userLoanAccount.getLoan(3000000.0);
            userLoanAccount.repayLoan(2000000.0);
            userLoanAccount.printStatement();

            System.out.println();

            CreditAccount userCreditAccount = new CreditAccount(name, 20000.0, 35000.0);
            userCreditAccount.getCredit(13000.0);
            userCreditAccount.repayCredit(6000.0);
            userCreditAccount.printStatement();

            scanner.close();
        } catch (InsufficientFundsException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
Enter your name: Deep
Enter initial balance: $450000
Amount: $600.0 deposited successfully!
Your balance is: $450600.0
Amount: $6000.0 withdrawn successfully
Your balance is: $444600.0
**************************************************
Bank Statement:
Account Number: 1
Account Holder's Name: Deep
Account balance: $444600.0
--------------------------------------------------

Not enough credit score.. Loan request declined!
No loans left to repay
**************************************************
Bank Statement:
Account Number: 2
Account Holder's Name: Deep
Account balance: $5000.0
--------------------------------------------------
Your credit score: 800.0
Your loan amount: $0.0
--------------------------------------------------

Credit granted worth: $13000.0
Repay was successful!!
Remaining repay amount: $8300.0
**************************************************
Bank Statement:
Account Number: 3
Account Holder's Name: Deep
Account balance: $20000.0
--------------------------------------------------
Your credit amount: $8300.0
--------------------------------------------------


...Program finished with exit code 0
Press ENTER to exit console.
```

SCREENSHOT: