1. Implement Stack data structure which will store objects of Contact class.

Contact class will have attributes : First name, Middle name, Last name, Date of birth, Gender, Mobile number. Follow Java Bean Specifications while designing Contact.java

Use exception handling to apply the following rules:

* 1. A pop must fail on an empty stack
  2. A push must fail on a stack which is already full
  3. A push must fail on a stack if the Contact object’s validation fails.

(Note : Stack class will have array of Contact objects as member variable and

It will have methods like pop and push. )

1. Create a class called AverageCalculator that has the following method:

public double calAverage(int n)

This method receives an integer as a parameter and calculates the average of first n natural numbers. If n is not a natural number, throw an exception **IllegalArgumentException** with an appropriate message.

1. Create a class BankAccount having the members as given below:

accno int

custname String

acctype String (indicates ‘Savings’ or ‘Current’)

balance float

Include the following methods in the BankAccount class:

void deposit(float amt);

void withdraw(float amt);

float getBalance();

deposit(float amt) method allows you to credit an amount into the current balance. If amount is negative, throw an exception InvalidAmountException to block the operation from being performed.

withdraw(float amt) method allows you to debit an amount from the current balance. Please ensure a minimum balance of Rs.1000/- in the account for savings account and Rs.5000/- for current account, else throw an exception InsufficientFunds and block the withdrawal operation.

Also throw an exception InvalidAmountException to block the operation from being performed if the amt parameter passed to this function is negative.

Add constructor in BankAccount to which you will pass, accno, custname, acctype and initial balance and check whether the balance is less than 1000 in case of savings account and less than 5000 in case of a current account. If so, then raise a **LowBalanceException**.

In either case if the balance is negative then raise the **InvalidAmountException** exception accordingly.

1. Create another class AccountChecker having main() method inside it which will create more than 5 different objects of BankAccount(reuse the class created in above assign) in an Array.

Provide method

boolean isValidAccount(int accno)

isValidAccount will accept the account no as input parameter

and check whether BankAccount with this account number exists

If not found then method should throw **AccountNotFoundException**