Concept : Exception Handling

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

```
public double avgFirstN(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.

Create a class Number having the following features:

Attributes

int	first number	
int	second number	
result	double	stores the result of arithmetic operations performed on a and b

Member functions

Number(x, y)	constructor to initialize the values of a and b
add()	stores the sum of a and b in result
sub()	stores difference of a and b in result
mul()	stores product in result
div()	stores a divided by b in result

Test to see if b is 0 and throw an appropriate exception since division by zero is undefined.

Display a menu to the user to perform the above four arithmetic operations.

Create a class BankAccount having the members as given below:

accNo integer
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 NegativeAmount 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. I 000/- 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 NegativeAmount to block the operation from being performed if the amt parameter passed to this function is negative.

getBalance() method returns the current balance. If the current balance is below the minimum required balance, then throw an exception LowBalanceException accordingly.

Have constructor to which you will pass, accno, cust_name, acctype and initial balance.
And check whether the balance is less than 1000 or not 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 NegativeAmount exception accordingly.

2. Create a class with following specifications.

Class Emp

empid	int
empName	string
designation	string
basic	double
hra	double readOnly

Methods
printDET()
calculateHRA()
printDET() methods will show details of the EMP.
calculateHRA() method will calculate HRA based on basic.

There will 3 designations supported by the application.

If designation is "Manager" - HRA will be 10% of BASIC
if designation is "Officer" - HRA will be 12% of BASIC if
category is "CLERK" - HRA will be 5% of BASIC

Have constructor to which you will pass, empid, designation, basic and price.

And checks whether the BASIC is less than 500 or not. If it is less than 500 raise a custom Exception as given below

Create LowSalException class with proper user message to handle BASIC less than 500.

3. Create The Custom Exceptions for Stack Full.and Stack Empty and throw them appropriately

4. Create The Custom Exceptions for Queue Full.and Queue Empty and throw them appropriately

5. Create a class Employee with id,name,salary. Ensure that id should be a non-negative integer, name should not be null
and max length should not exceed 30, salary should be non-negative and max value permitted
is 99999/-
Hint : Try creating Exception Classes as
NegativeValueException
MaxLengthExceededException
MaxValueExceededException