

Name : Shreyash Shete
Batch : May 2022

Core Java Assignments

Exception Handling

Concept : Exception Handling

Estimated time: 2 Hours

- 1) Bank account exception handling :

Class Account :

```
public class Account {    // Code for Exception handling...

    private int accNo;
    private String custName;
    private double accBal;

    public Account(int accNo, String custName, double accBal) {
        super();
        this.accNo = accNo;
        this.custName = custName;
        this.accBal = accBal;
    }

    public int getAccNo() {
        return accNo;
    }

    public void setAccNo(int accNo) {
        this.accNo = accNo;
    }

    public String getCustName() {
```

```

        return custName;
    }

    public void setCustName(String custName) {
        this.custName = custName;
    }

    public double getAccBal() {
        return accBal;
    }

    public void setAccBal(double accBal) {
        this.accBal = accBal;
    }
}

```

Class AccountInfo :

```

import java.util.Scanner;

public class AccountInfo {

    public Account[] createAccount() throws BalanceException {
        // to create the account

        @SuppressWarnings("resource")
        Scanner sc = new Scanner(System.in);

        System.out.println("How many Accounts do you want ?");

        int n = sc.nextInt();

        Account arr[] = new Account[n];
    }
}

```

```

    for(int i=0;i<arr.length;i++)
    {
        System.out.println("Enter Account number : ");
        int accno = sc.nextInt();
        System.out.println("Enter Customer name : ");
        String custn = sc.next();
        System.out.println("Enter Account balance : ");
        double accbal = sc.nextDouble();

        try {                                //Exception Handling..
            if(accbal < 1000) {
                throw new BalanceException();
            }
        }
        catch(BalanceException b) {
            System.err.println("Exception : " + b);
            throw b;
        }

        Account aobj = new Account(accno, custn, accbal);
        arr[i] = aobj;
    }
    return arr;
}

public void displayAccount(Account arr[]) { // to display
the account.,,

    System.out.println("\n-----Details of the account----
-----");
    for(Account a : arr)
    {
        System.out.println(a.getAccNo() + "\t" +
a.getCustName() + "\t" + a.getAccBal());
    }
}

public Account searchAccount(Account arr[], int ano) {
    Account temp=null;

    for(Account a : arr)
    {
        if(a.getAccNo()==ano) {
            temp = a;

```

```

        break;
    }
}
return temp;
}

public void deposit(Account temp, double damt) {
    temp.setAccBal(temp.getAccBal()+ damt);
    System.out.println("Amount deposited !");
    System.out.println("New amount is " +
temp.getAccBal());
}

public void withdraw(Account temp, double amt) {
    if(temp.getAccBal()> amt)
    {
        temp.setAccBal(temp.getAccBal() - amt);
        System.out.println("Amount can be withdrawn"); //
to check whether the given amount is greater than the amount
which is already present in the balance
    }
    else
    {
        System.out.println("Amount cannot be withdrawn
");
    }
    System.out.println("New amount is " +
temp.getAccBal());
}

public void sortAccount(Account arr[]) { // for sorting
    Account temp=null;

    System.out.println("-----Sorting by Account balance-----
----");
    for(int i=0;i<arr.length;i++)
    {
        for(int j=(i+1);j<arr.length;j++)
        {
            if(arr[i].getAccBal() > arr[j].getAccBal())
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

```

```

    }
}
}

```

Class AccountMain :

```

import java.util.Scanner;

public class AccountMain {

    public static void main(String[] args) throws
    BalanceException {

        @SuppressWarnings("resource")
        Scanner sc = new Scanner(System.in);
        Account arr[] = null;
        AccountInfo ainfo = new AccountInfo();

        int ch = 0;

        do
        {
            System.out.println("\n1 - Create Account\n2 -
Display Account\n3 - Search Account\n4 - Sorting");
            System.out.println("Enter your choice : ");
            ch = sc.nextInt();

            switch(ch)
            {
                case 1 :
                    try {
                        arr = ainfo.createAccount();
                    }
                    catch(Exception e) {
                        System.out.println("Exception : " +
e);

                        e.printStackTrace();
                    }
                    break;

                case 2 :
                    ainfo.displayAccount(arr);
                    break;
            }
        }
    }
}

```

```

        case 3 :
            System.out.println("Enter Account number do
you want to search in the record : ");
            int ano = sc.nextInt();
            Account temp = ainfo.searchAccount(arr,
ano);
            if(temp!=null)
            {
                System.out.println("\nRecord found
\n");
                System.out.println(temp.getAccNo() +
"\t" + temp.getCustName() + "\t" + temp.getAccBal());

                System.out.println("Do you want to
preformed transaction ? y/n ");
                String str = sc.next();
                int cho = 0;

                if(str.equals("y"))
                {
                    System.out.println("1 -
Deposit\n2 - Withdraw");
                    System.out.println("Enter your
choice : ");
                    cho=sc.nextInt();

                    if(cho==1)
                    {
                        System.out.println("Enter
amount to deposit : ");
                        sc.nextDouble();

                        System.out.println("1 -
Deposit\n2 - Withdraw");
                        System.out.println("Enter your
choice : ");
                        cho=sc.nextInt();

                        if(cho==1)
                        {
                            System.out.println("Enter
amount to deposit : ");
                            sc.nextDouble();

                            ainfo.deposit(temp, damt);
                        }
                        else if(cho==2)
                        {
                            System.out.println("Enter
amount to withdraw : ");
                            sc.nextDouble();

                            ainfo.withdraw(temp, amt);
                            break;
                        }
                    }
                }
                if(str.equals("n"))
                {

```

```

        System.out.println("Transaction completed");
    }
}
else
{
    System.out.println("\nRecord not found
\n");
    break;
}

case 4 :
    ainfo.sortAccount(arr);
    ainfo.displayAccount(arr);
    // sorting command
    break;

default :
    System.out.println("Invalid choice ");
    break;
}
System.out.println("\nDo you want to continue?
press1 \n");
ch = sc.nextInt();
}
while(ch==1);
System.out.println("-----Thank you-----");
}
}

```

Class BalanceException :

```

public class BalanceException extends Exception {

    public String toString() {
        return "Balance must be greater than 1000";
    }

}

```

2) Shopping Application Exception :

Class Address :

```
public class Address {  
  
    private String custCity;  
    private String custState;  
    private String custPin;  
  
    public Address(String custCity, String custState, String  
custPin) {  
        super();  
        this.custCity = custCity;  
        this.custState = custState;  
        this.custPin = custPin;  
    }  
  
    public String getCustCity() {  
        return custCity;  
    }  
  
    public void setCustCity(String custCity) {  
        this.custCity = custCity;  
    }  
  
    public String getCustState() {  
        return custState;  
    }  
  
    public void setCustState(String custState) {  
        this.custState = custState;  
    }  
  
    public String getCustPin() {  
        return custPin;  
    }  
  
    public void setCustPin(String custPin) {  
        this.custPin = custPin;  
    }  
}
```


Class Customer :

```
public class Customer {

    private int custId;
    private String custName;
    private String custMob;
    private Product prod;
    private Address add;
    private Product produ[];
    private Customer custom[];

    public Customer(int custId, String custName, String custMob,
Product prod, Address add, Product[] produ,
Customer[] custom) { // Constructor for many
customer many product ....
        super();
        this.custId = custId;
        this.custName = custName;
        this.custMob = custMob;
        this.prod = prod;
        this.add = add;
        this.produ = produ;
        this.custom = custom;
    }

    public Customer(int custId, String custName, String custMob,
Address add, Product[] produ) { // Constructor for product array
ie one customer multiple products...
        super();
        this.custId = custId;
        this.custName = custName;
        this.custMob = custMob;
        this.add = add;
        this.produ = produ;
    }

    public Customer(int custId, String custName, String custMob,
Product prod, Address add) { //Constructor for product class
        super();
        this.custId = custId;
        this.custName = custName;
        this.custMob = custMob;
    }
}
```

```
        this.prod = prod;
        this.add = add;
    }
```

```
    public Customer(int i, Object object, Object object2,
Address add2, Product product) {
```

```
    }
```

```
    public int getCustId() {
        return custId;
    }
```

```
    public void setCustId(int custId) {
        this.custId = custId;
    }
```

```
    public String getCustName() {
        return custName;
    }
```

```
    public void setCustName(String custName) {
        this.custName = custName;
    }
```

```
    public String getCustMob() {
        return custMob;
    }
```

```
    public void setCustMob(String custMob) {
        this.custMob = custMob;
    }
```

```
    public Product getProd() {
        return prod;
    }
```

```
    public void setProd(Product prod) {
        this.prod = prod;
    }
```

```
    public Address getAdd() {
        return add;
    }
```

```

    }

    public void setAdd(Address add) {
        this.add = add;
    }

    public Product[] getProdu() {
        return produ;
    }

    public void setProdu(Product[] produ) {
        this.produ = produ;
    }

    public Customer[] getCustom() {
        return custom;
    }

    public void setCustom(Customer[] custom) {
        this.custom = custom;
    }
}

```

Class Product :

```

public class Product {

    private int prodId;
    private int prodqty;
    private String prodName;
    private double prodPrice;

    public Product(int prodId, int prodqty, String prodName,
double prodPrice) {
        super();
    }
}

```

```
        this.prodId = prodId;
        this.prodqty = prodqty;
        this.prodName = prodName;
        this.prodPrice = prodPrice;
    }

    public int getProdId() {
        return prodId;
    }

    public void setProdId(int prodId) {
        this.prodId = prodId;
    }

    public int getProdqty() {
        return prodqty;
    }

    public void setProdqty(int prodqty) {
        this.prodqty = prodqty;
    }

    public String getProdName() {
        return prodName;
    }

    public void setProdName(String prodName) {
        this.prodName = prodName;
    }

    public double getProdPrice() {
        return prodPrice;
    }

    public void setProdPrice(double prodPrice) {
        this.prodPrice = prodPrice;
    }
}
```

Class ShopInfo :

```
import java.util.Scanner;

public class ShopInfo {

    Scanner sc = new Scanner(System.in);

    public Customer create() { // Create Single Product

        System.out.println("-----Program to display
single product, single customer-----\n");
        System.out.println("Enter Product Id, Product
quantity, Product name, Product price : ");
        Product prod = new Product(sc.nextInt(), sc.nextInt(),
sc.next(), sc.nextDouble());

        System.out.println("Enter Customer City, Customer
State, Customer Pincode : ");
        Address add = new Address(sc.next(), sc.next(),
sc.next());

        System.out.println("Enter Customer Id, Customer Name,
Customer Mobile no : ");
        int id = sc.nextInt();
        String name = sc.next();
        String mob = sc.next();
        Customer cust = new Customer(id, name, mob, prod,
add);

        return cust;
    }

    public Customer createMul() throws ProductException { //
Create Single Customer, Multiple products...

        System.out.println("-----To create Multiple
products, one customer -----\n");
        System.out.println("How many product do you want : ");
        int n = sc.nextInt();

        Product produ[] = new Product[n];
```

```

        for(int i=0;i<produ.length;i++)
        {
            System.out.println("Enter product Id, Product
quantity, Product name and Product price : ");
            int prodid = sc.nextInt();
            int prodqt = sc.nextInt();
            String prodname = sc.next();
            double prodPrice = sc.nextDouble();

            try {
                if(prodPrice < 50) {
                    throw new ProductException();
                }
            }
            catch(ProductException p) {
                System.err.println("Exception : " + p);
                throw p;
            }

            Product prod = new Product(prodid, prodqt,
prodname, prodPrice);
            produ[i] = prod;
        }

        System.out.println("Enter Customer City, Customer
State, Customer Pincode : ");
        Address add = new Address(sc.next(), sc.next(),
sc.next());

        System.out.println("Enter Customer Id, Customer Name,
Customer Mobile no : ");
        int custid = sc.nextInt();
        String custname = sc.next();
        String custMob = sc.next();
        Customer cust = new Customer(custid, custname,
custMob, add, produ);

        return cust;
    }

    public Customer[] Manycust() { //Multiple customer, multiple
products..

```

```

        System.out.println("-----To create Multiple
products, Multiple customer -----\n");
        System.out.println("How many Customer do you want :
");
        int n = sc.nextInt();

        Customer cust[] = new Customer[n];
        for(int i=0;i<cust.length;i++)
        {
            System.out.println("Enter Customer City, Customer
State, Customer Pincode : ");
            Address add = new Address(sc.next(), sc.next(),
sc.next());

            System.out.println("Enter Customer Id, Customer
Name, Customer Mobile no : ");
            int custid = sc.nextInt();
            String custname = sc.next();
            String custMob = sc.next();

            System.out.println("How many product do you want
: ");
            int n1 = sc.nextInt();

            Product produ[] = new Product[n1];
            for(int i1=0;i1<produ.length;i1++)
            {
                System.out.println("Enter product Id,
Product quantity, Product name and Product price : ");
                int prodid = sc.nextInt();
                int prodqt = sc.nextInt();
                String prodname = sc.next();
                double prodPrice = sc.nextDouble();

                Customer cust1 = new Customer(custid,
custname, custMob, add, produ);
                Product prod = new Product(prodid, prodqt,
prodname, prodPrice);
                produ[i1] = prod;
            }
        }

        return cust ;

```

```

    }

    // Display functions....

    public void display(Customer cust) { //1
        System.out.println("-----Customer Details-----
        -----");
        System.out.println("Customer ID :- " +
        cust.getCustId());
        System.out.println("Customer Name :- " +
        cust.getCustName());
        System.out.println("Customer Mobile No. :- " +
        cust.getCustMob());

        System.out.println("-----Product Details-----
        -----");
        System.out.println("Product Id :- " +
        cust.getProd().getProdId());
        System.out.println("Product quantity :- " +
        cust.getProd().getProdQty());
        System.out.println("Product Name :- " +
        cust.getProd().getProdName());
        System.out.println("Product price :- " +
        cust.getProd().getProdPrice());

        System.out.println("-----Customer Address--
        -----");
        System.out.println("Customer City :- " +
        cust.getAdd().getCustCity());
        System.out.println("Customer State :- " +
        cust.getAdd().getCustState());
        System.out.println("Customer Pincode :- " +
        cust.getAdd().getCustPin());

    }

    public void displayArr(Customer cust) { //2
        System.out.println("-----Customer Details-----
        -----");
        System.out.println("Customer ID :- " +
        cust.getCustId());

```



```

        System.out.println("Customer Name :- " +
cust.getCustName());
        System.out.println("Customer Mobile No. :- " +
cust.getCustMob());

        System.out.println("-----Product Details-----
-----");
        Product produ[] = cust.getProdu();
        for(Product p : produ)
        {
            System.out.println("Product Id :- " +
p.getProdId());
            System.out.println("Product quantity :- " +
p.getProdqty());
            System.out.println("Product Name :- " +
p.getProdName());
            System.out.println("Product price :- " +
p.getProdPrice());

        }

        System.out.println("-----Customer Address--
-----");
        System.out.println("Customer City :- " +
cust.getAdd().getCustCity());
        System.out.println("Customer State :- " +
cust.getAdd().getCustState());
        System.out.println("Customer Pincode :- " +
cust.getAdd().getCustPin());
    }

```

```

    public void displayManycust(Customer cust[], Customer[]
prod2) { //3
        System.out.println("----- Customer Details And
Address -----");
        for(int i=0;i<cust.length;i++) {
            System.out.println("Customer ID :- " +
cust[i].getCustId());
            System.out.println("Customer Name :- " +
cust[i].getCustName());
            System.out.println("Customer Mobile No :- " +
cust[i].getCustMob());
            System.out.println("Customer City : " +
cust[i].getAdd().getCustCity());

```

```

        System.out.println("Customer State : " +
cust[i].getAdd().getCustState());
        System.out.println("Customer Pin : " +
cust[i].getAdd().getCustPin());

        for(int j=0;j<prod2.length;j++) {
            System.out.println("Product ID : " +
prod2[j].getProd().getProdId());
            System.out.println("Product Quantity : " +
prod2[j].getProd().getProdQty());
            System.out.println("Product Name : " +
prod2[j].getProd().getProdName());
            System.out.println("Product price : " +
prod2[j].getProd().getProdPrice());
        }
    }
}

```

Class ShopMain :

```

import java.util.Scanner;

public class ShopMain {

    public static void main(String[] args) {

        @SuppressWarnings("resource")
        Scanner sc = new Scanner(System.in);
        int ch = 0;
        Address add = new Address(null, null, null);
        Product product = new Product(0, 0, null, 0);
        Customer customer = new Customer(0, null, null, add,
product);
        ShopInfo sinfo = new ShopInfo();

        do

```

```

        {
            System.out.println("1 - To display Customer Bill
with single product data\n2 - To display Customer Bill with
multiple product data\n3 - To display Multiple customer bill With
multiple product data");
            System.out.println("Enter your Choice : ");
            ch = sc.nextInt();

            switch(ch)
            {
                case 1 :
                    Customer cust = sinfo.create();
                    sinfo.display(cust);
                    break;

                case 2 :
                    try {
                        Customer cust1 = sinfo.createMul();
                        sinfo.displayArr(cust1);
                    }
                    catch(Exception e) {
                        System.out.println("Exception : " +
e);
                        e.printStackTrace();
                    }

                    break;

                case 3 :
                    Customer[] cust2 = sinfo.Manycust();
                    Customer[] prod2 = sinfo.Manycust();
                    sinfo.displayManycust(cust2, prod2);

                    break;

                default :
                    System.out.println("-----Invalid Choice----
--");
                    break;
            }
            System.out.println("\nDo you want to continue ?
(Press 1 to continue) \n");
            ch = sc.nextInt();
        }
        while(ch==1);
    }
}

```

```
        System.out.println("-----Thank you-----  
-");  
    }  
}
```

Class ProductException:

```
public class ProductException extends Exception {  
    public String toString() {  
        return "Product Prize must be atleast greater than  
50";  
    }  
}
```

3. 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.

Class Number :

```
import java.util.Scanner;

public class Number {

    Scanner sc = new Scanner(System.in);
    private int firstNo;
    private int secondNo;

    public Number(int firstNo, int secondNo) { // generate
constructor..
        super();
        this.firstNo = firstNo;
        this.secondNo = secondNo;
    }
```

```

    public int getFirstNo() { // getter setter..
        return firstNo;
    }
    public void setFirstNo(int firstNo) {
        this.firstNo = firstNo;
    }
    public int getSecondNo() {
        return secondNo;
    }
    public void setSecondNo(int secondNo) {
        this.secondNo = secondNo;
    }

    public void add() { //addition

        System.out.println("Enter any 2 numbers..");
        int firstNo = sc.nextInt();
        int secondNo = sc.nextInt();

        double result = firstNo + secondNo;
        System.out.println("Addition of 2 numbers is : " +
result);
    }

    public void sub() { //subtraction

        System.out.println("Enter any 2 numbers..");
        int firstNo = sc.nextInt();
        int secondNo = sc.nextInt();

        double result = firstNo - secondNo;
        System.out.println("Subtraction of 2 numbers is : " +
result);
    }

    public void mul() { //Multiplication

        System.out.println("Enter any 2 numbers..");
        int firstNo = sc.nextInt();
        int secondNo = sc.nextInt();

        double result = firstNo * secondNo;

```

```

        System.out.println("Multiplication of 2 numbers is : "
+ result);
    }

    public void div() throws NumberException { //division...

        System.out.println("Enter any 2 numbers..");
        int firstNo = sc.nextInt();
        int secondNo = sc.nextInt();

        try {
            if(secondNo == 0) {
                throw new NumberException();
            }
        }
        catch(NumberException ex) {
            System.err.println("Exception is " + ex);
            throw ex;
        }

        double result = firstNo / secondNo;
        System.out.println("Subtraction of 2 numbers is : " +
result);
    }

}

Class CalcAvg :

```

```

public class CalcAvg {
    int c;

    public double avgFirstN(int N)
    {
        if(N<0)
        {
            try{
                throw new NumberFormatException();
            }
        }
    }
}

```

```

        catch(NumberFormatException nf){
            System.out.println("We Must To Enter The P0sitive
Number");
        }
        catch(Exception e) {
            System.out.println("Some other Exception.." + e);
        }

    }
    else
    {
        int c=N*10;

        System.out.println("The Average is a = "+c);
    }

    return N;
}
}

```

Class TestNumber :

```

import java.util.Scanner;

public class TestNumber {

    public static void main(String[] args) {

        Number no = new Number(0, 0);
        Scanner sc = new Scanner(System.in);
        int ch =0;

        do {

```



```
System.out.println("1 - To print the addition of  
2 numbers..\n2 - To print the subtraction of 2 numbers...\n3 - To  
print the multiplication of 2 numbers..\n4 - To print the  
Division of 2 numbers..\n5 - To print the Average of natural  
numbers.. ");
```

```
System.out.println("Enter your choice : ");  
ch = sc.nextInt();
```

```
switch (ch)
```

```
{
```

```
case 1 :
```

```
no.add();
```

```
break;
```

```
case 2 :
```

```
no.sub();
```

```
break;
```

```
case 3 :
```

```
no.mul();
```

```
break;
```

```
case 4 :
```

```
try {
```

```
no.div();
```

```
}
```

```
catch(Exception e) {
```

```
System.out.println("Exception : " +
```

```
e);
```

```
e.printStackTrace();
```

```
}
```

```
break;
```

```
case 5 :
```

```
CalcAvg t=new CalcAvg();
```

```
System.out.println("Enter any Natural  
number do you want to find average..");
```

```
int n = sc.nextInt();
```

```
t.avgFirstN(n);
```

```
break;
```

```
default :
```

```
System.out.println("Invalid choice!");
```

```
break;
```

```
}
```

```

        System.out.println("Do you want to continue ?
press 1");
        ch = sc.nextInt();

        }while(ch==1);
        System.out.println("-----Thank you-----");
    }
}

```

Class NumberException :

```

public class NumberException extends Exception {

    public String toString() {
        return "Division by zero is not possible..";
    }

}

```

4) 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. 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 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.

Class BankAccount :

```
public class BankAccount {

    private int accNo;
    private String custName;
    private String accType;
    private double balance;

    public BankAccount(int accNo, String custName, String
accType, double balance) {
        super();
        this.accNo = accNo;
        this.custName = custName;
        this.accType = accType;
        this.balance = balance;
    }

    public int getAccNo() {
        return accNo;
    }
    public void setAccNo(int accNo) {
        this.accNo = accNo;
    }
    public String getCustName() {
        return custName;
    }
    public void setCustName(String custName) {
        this.custName = custName;
    }
    public String getAccType() {
        return accType;
    }
    public void setAccType(String accType) {
        this.accType = accType;
    }
    public double getBalance() {
        return balance;
    }
    public void setBalance(double balance) {
        this.balance = balance;
    }
}
```

Class AccountInfo :

```
import java.util.Scanner;

public class AccountInfo {
    Scanner sc = new Scanner(System.in);

    public BankAccount[] createAccount() throws AccountException
    { //to create the account

        System.out.println("How many Accounts do you want :
");
        int n = sc.nextInt();

        BankAccount arr[] = new BankAccount[n];
        for(int i=0;i<arr.length;i++) {
            System.out.println("Enter Account number : ");
            int accno = sc.nextInt();
            System.out.println("Enter Customer name : ");
            String custn = sc.next();
            System.out.println("Enter Account type
(Savings/current) : ");
            String acctype = sc.next();
            System.out.println("Enter Account balance : ");
            double accbal = sc.nextDouble();

            try {
                if(accbal < 0) {
                    throw new AccountException();
                }
            }
            catch(AccountException ex) {
                System.err.println("Exception : " + ex);
                throw ex;
            }

            BankAccount obj = new BankAccount(accno, custn,
acctype, accbal);
            arr[i] = obj;

        }
        return arr;
    }
}
```

```

    public void displayAccount(BankAccount arr[]) throws
EmptyAccount{ //To display the account

        System.out.println("----Customer Details-----");
        for(int i=0;i<arr.length;i++) {

            try {
                if(arr.length == 0) {
                    throw new EmptyAccount();
                }
            }
            catch (EmptyAccount ex) {
                System.err.println("Exception : " + ex);
                throw ex;
            }

            System.out.println("Account NO : " +
arr[i].getAccNo());
            System.out.println("Customer Name : " +
arr[i].getCustName());
            System.out.println("Account type : " +
arr[i].getAccType());
            System.out.println("Account balance : " +
arr[i].getBalance());
        }

    }

    public BankAccount search(BankAccount arr[], int ano) { //
to search the account
        BankAccount temp = null;

        for(BankAccount a : arr)
        {
            if(a.getAccNo()==ano) {
                temp = a;
                break;
            }
        }
        return temp;
    }

```

```

        public void deposit(BankAccount temp, double damt) {
//deposit
            temp.setBalance(temp.getBalance() + damt);
            System.out.println("Amount Deposited !");
            System.out.println("New amount is : " +
temp.getBalance());
        }

        public void withdraw(BankAccount temp, double amt) {
            if(temp.getBalance() > amt) {
                temp.setBalance(temp.getBalance() - amt);
                System.out.println("Amount can be withdrawn ");
            }
            else {
                System.out.println("Amount cannot be withdrawn
");
            }
            System.out.println("New amount is : " +
temp.getBalance());
        }

    }

```

Class BankMain :

```
import java.util.Scanner;

public class BankMain {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        BankAccount arr[] = null;
        AccountInfo ainfo = new AccountInfo();
        int ch = 0;

        do {
            System.out.println("\n1 - Create Account\n2 -
Display Account\n3 - Search Account");
            System.out.println("Enter your choice : ");
            ch = sc.nextInt();

            switch(ch)
            {
                case 1 :
                    try {
                        arr = ainfo.createAccount();
                    } catch (Exception e) {
                        System.out.println("Exception : " +
e);
                        e.printStackTrace();
                    }
                    break;

                case 2 :
                    try {
                        ainfo.displayAccount(arr);
                    } catch (Exception e) {
                        System.out.println("Exception : " +
e);
                        e.printStackTrace();
                    }
                    break;

                case 3 :
```



```

        System.out.println("Enter Account number do
you want to search in the record : ");
        int ano = sc.nextInt();
        BankAccount temp = ainfo.search(arr, ano);
        if(temp!=null)
        {
            System.out.println("\nRecord found
\n");
            System.out.println(temp.getAccNo() +
"\t" + temp.getCustName() + "\t" + temp.getAccType()+ "\t" +
temp.getAccNo());

            System.out.println("Do you want to
preformed transaction ? y/n ");
            String str = sc.next();
            int cho = 0;

            if(str.equals("y"))
            {
                System.out.println("1 -
Deposit\n2 - Withdraw");
                System.out.println("Enter your
choice : ");
                cho=sc.nextInt();

                if(cho==1)
                {
                    System.out.println("Enter
amount to deposit : ");
                    sc.nextDouble();

                    double damt =
ainfo.deposit(temp, damt);
                }
                else if(cho==2)
                {
                    System.out.println("Enter
amount to withdraw : ");
                    sc.nextDouble();

                    double amt =
ainfo.withdraw(temp, amt);
                    break;
                }
            }
            if(str.equals("n"))
            {
                System.out.println("Transaction completed");
            }
        }
    }
}

```

```

        }
    }
}
else
{
    System.out.println("\nRecord not found
\n");
    break;
}

default :
    System.out.println("Invalid choice ");
    break;

}System.out.println("\nDo you want to continue?
press1 \n");
ch = sc.nextInt();

}while(ch==1);
System.out.println("-----Thank you-----");
}
}

```

Class AccountException :

```
public class AccountException extends Exception {  
    public String toString() {  
        return "Balance cannot be negative";  
    }  
}
```

Class EmptyAccount :

```
public class EmptyAccount extends Exception {  
    public String toString() {  
        return "No data in the account";  
    }  
}
```

5. 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.

Class Emp :

```
public class Emp {  
  
    private int empId;  
    private String empName;  
    private String designation;  
    private double basic;  
    private double hra;  
  
    public Emp(int empId, String empName, String designation,  
double basic, double hra) {  
        super();  
        this.empId = empId;  
        this.empName = empName;  
        this.designation = designation;  
        this.basic = basic;  
        this.hra = hra;  
    }  
  
    public int getEmpId() {  
        return empId;  
    }  
  
    public void setEmpId(int empId) {  
        this.empId = empId;  
    }  
  
    public String getEmpName() {  
        return empName;  
    }  
  
    public void setEmpName(String empName) {  
        this.empName = empName;  
    }  
  
    public String getDesignation() {  
        return designation;  
    }  
  
    public void setDesignation(String designation) {  
        this.designation = designation;  
    }  
}
```

```
    public double getBasic() {  
        return basic;  
    }  
  
    public void setBasic(double basic) {  
        this.basic = basic;  
    }  
  
    public double getHra() {  
        return hra;  
    }  
  
    public void setHra(double hra) {  
        this.hra = hra;  
    }  
  
}
```

Class EmpInfo :

```
import java.util.Scanner;

public class EmpInfo {
    Scanner sc = new Scanner(System.in);

    public Emp[] printDet() throws LowSalException {

        System.out.println("How many Accounts do you want : ");

        int n = sc.nextInt();

        Emp arr[] = new Emp[n];
        for(int i=0;i<arr.length;i++) {
            System.out.println("Enter Employee ID : ");
            int eid = sc.nextInt();
            System.out.println("Enter Employee Name : ");
            String ename = sc.next();
            System.out.println("Enter Employee Designation : ");

            String edeg = sc.next();
            System.out.println("Enter Basic Salary of Employee : ");

            double basic = sc.nextDouble();

            try {
                if(basic < 500) {
                    throw new LowSalException();
                }
            }
            catch(LowSalException ex) {
                System.err.println("Exception : " + ex);
                throw ex;
            }

            System.out.println("Enter HRA of Employee : ");
            double hra = sc.nextDouble();

            Emp obj = new Emp(eid, ename, edeg, basic, hra);
            arr[i] = obj;
        }
        return arr;
    }
}
```

```

    }

    public void calculateHra(Emp arr[]) throws Blankdata {

        System.out.println("---Employee Details---");

        for(int i=0;i<arr.length;i++) {

            try { // exception for empty account
                if(arr.length==0) {
                    throw new Blankdata();
                }
            }
            catch(Blankdata ex) {
                System.err.println("Exception : " + ex);
                throw ex;
            }

            System.out.println("Employee ID : " +
arr[i].getEmpId());
            System.out.println("Employee Name : " +
arr[i].getEmpName());
            System.out.println("Employee Designation : " +
arr[i].getDesignation());

            if(arr[i].getDesignation() == "manager") {
                double temp = arr[i].getHra() * 0.1;
                System.out.println("Employee basic Salary
is : " + temp);
            }
            else if(arr[i].getDesignation() == "officer") {
                double temp = arr[i].getHra() * 0.12;
                System.out.println("Employee basic Salary
is : " + temp);
            }
            else if(arr[i].getDesignation() == "clerk") {
                double temp = arr[i].getHra() * 0.05;
                System.out.println("Employee basic Salary
is : " + temp);
            }
        }
    }
}

```



```
        System.out.println("Employee Hra is : " +  
arr[i].getHra());  
    }  
}  
}
```

Class EmpMain :

```
import java.util.Scanner;

public class EmpMain {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Emp arr[] = null;
        EmpInfo einfo = new EmpInfo();
        int ch = 0;

        do
        {
            System.out.println("\n1 - Create Employee data\n2
- Display Employee data");
            System.out.println("\nEnter your choice : ");
            ch = sc.nextInt();

            switch(ch)
            {
                case 1 :

                    try {
                        arr = einfo.printDet();
                    } catch (Exception e) {
                        System.out.println("Exception : " +
e);
                        e.printStackTrace();
                    }
                    break;

                case 2 :
                    try {
                        einfo.calculateHra(arr);
                    } catch (Exception e) {
                        System.out.println("Exception : " +
e);
                        e.printStackTrace();
                    }
                    break;

                default :
                    System.out.println("Invalid choice !");
            }
        }
    }
}
```

```

        break;
    }

    System.out.println("Do you want to continue?
press1");
    ch = sc.nextInt();

    }while(ch==1);
    System.out.println("----Thank You-----");
}
}

```

Class LowSalException :

```

public class LowSalException extends Exception {

    public String toString() {
        return "Employee Salary must be greater than 500";
    }

}

```

Class Blankdata :

```

public class Blankdata extends Exception {

    public String toString() {
        return "No Employee data present";
    }

}

```

6. Create a class USERTRAIL with following specifications.
val1, val2 type int

Methods

boolean show () will check if val1 and val2 are greater or less than Zero
have constructor which will val1, val2 and check whether if it is less than 0 then
raise a custom Exception (name: Illegal value exception.)

Class Usertrail :

```
public class Usertrail {  
  
    private int val1;  
    private int val2;  
  
    public Usertrail(int val1, int val2) {  
        super();  
        this.val1 = val1;  
        this.val2 = val2;  
    }  
  
    public int getVal1() {  
        return val1;  
    }  
  
    public void setVal1(int val1) {  
        this.val1 = val1;  
    }  
  
    public int getVal2() {  
        return val2;  
    }  
  
    public void setVal2(int val2) {  
        this.val2 = val2;  
    }  
}
```

Class trailInfo :

```
import java.util.Scanner;

public class trailInfo {
    Scanner sc = new Scanner(System.in);

    public void check() throws ValueException {

        Usertrail u = new Usertrail(0, 0);
        System.out.println("Enter any 2 numbers..");
        int val1 = sc.nextInt();
        int val2 = sc.nextInt();

        try {
            if( val1 == 0 || val2 == 0) {
                throw new ValueException();
            }
        }
        catch(ValueException ex) {
            System.err.println("Exception : " + ex);
            throw ex;
        }
    }
}
```

Class ValueMain :

```
import java.util.Scanner;

public class ValueMain {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        trailInfo tinfo = new trailInfo();
        int ch = 0;

        do
```

```

        {
            try {
                tinfo.check();
            } catch (Exception e) {
                System.out.println("Exception : " + e);
                e.printStackTrace();
            }

            System.out.println("Do you want to continue?
press1");

            ch = sc.nextInt();

        }while(ch==1);
        System.out.println("-----Thank you-----");
    }
}

```

Class ValueException :

```

public class ValueException extends Exception {

    public String toString() {
        return "Value must be greater than zero";
    }

}

```