1. Problem Statement: Develop a BankAccount class that implements core banking operations:

- balanceEnquiry(): Displays the current account balance.
- withdraw(): Deducts the specified amount from the account balance.
- deposit(): Adds the specified amount to the account balance.

Implement user-defined exceptions:
a. LowBalanceException: Thrown when a withdrawal amount exceeds the available balance.
b. NegativeNumberException: Thrown when attempting to deposit or withdraw a negative amount.

```
import java.lang.*;
import java.util.*;

class LowBalanceException extends Exception{
 public String toString(){
  return "You are trying to withdraw amount more than available amount";
  }
}
class NegativeNumberException extends Exception{
 public String toString(){
  return "You can't withdraw or deposit negative amount so enter positive amount";
  }
}

class BankAccount{

 Scanner sc = new Scanner(System.in);
  double currentAmount, get, give;

 void balanceEnquiry(){
  System.out.println("Enter your current balance: ");
  currentAmount = sc.nextDouble();
  System.out.println("Your current balance is: "+currentAmount);
  }

  void withdraw(){
  try{
  System.out.println("Enter the amount you want to withdraw : ");
  get = sc.nextDouble();
  if(get > currentAmount){
  throw new LowBalanceException();
  }
  else if(get < 0){
  throw new NegativeNumberException();
  }
  else{
  double c = currentAmount - get;
  System.out.println("Your current balance is: "+c);
  }
  }
  catch(LowBalanceException e){
  System.out.println(e);
   }
   catch(NegativeNumberException n){
   System.out.println(n);
   }
  }
```

```
  void deposit(){
  try{

  System.out.println("Enter the amount you want to deposit : ");
  give = sc.nextDouble();
  if(give < 0){
  throw new NegativeNumberException();
  }
  else{
  double d = currentAmount + give;
  System.out.println("Your current balance is: "+d);
  }
  }
  catch(NegativeNumberException e){
  System.out.println(e);}
  }

  public static void main(String[] ar){
  Scanner sc = new Scanner(System.in);
  BankAccount a = new BankAccount();
  while(true){
   System.out.println("\nChoose an option: ");
   System.out.println("\n1.Current Balance\n2.withdraw\n3.deposit\n4.exit");
   int choice = sc.nextInt();
   switch(choice){
   case 1:
   a.balanceEnquiry();
   break;
   case 2:
   a.withdraw();
   break;
   case 3:
   a.deposit();
   break;
   case 4:
   System.exit(4);
  }
  }
  }
}
```

Choose an option:
1.Current Balance
2.withdraw
3.deposit
4.exit
1
Enter your current balance:
1200
Your current balance is: 1200.0

Choose an option:
1.Current Balance
2.withdraw
3.deposit
4.exit
2
Enter the amount you want to withdraw :
1250
You are trying to withdraw amount more than available amount

---

2. Write a Java program with a method that takes an integer as input. If the number is odd, the method should throw a custom exception (OddNumberException). Handle this exception in the main program.

```
import java.lang.*;
import java.util.*;

class OddNumberException extends Exception{
 public String toString(){
   return "Number is odd";
  }
}

class Odd{
public static void main(String[] ar){

 try{
  Scanner sc = new Scanner(System.in);
 System.out.println("Enter a number: ");
 int num = sc.nextInt();
 if(num % 2 != 0){
   throw new OddNumberException();
   }
 else{
  System.out.println("Entered number is even");
}
}
catch(OddNumberException e){
 System.out.println(e);
 }
  }
}
```

Output:

Enter a number:
12
Entered number is even

3. Create a package ExceptionHandlingDemo containing classes Calculator and DivisionException.

- The Calculator class should have a method divide(int a, int b) that performs division.
- If b is zero, throw a custom exception DivisionException with an appropriate error message.
- Handle the exception in the main program and display an error message instead of crashing.If the number is odd, the method should throw a custom exception (OddNumberException). Handle this exception in the main program.

```
package ExceptionHandlingDemo;

public class Calculator{
 public void divide(int a, int b){
  System.out.println("the answer is: "+(a / b));
  }
}
```

```
package ExceptionHandlingDemo;

public class DivisionException extends Exception{
   public String toString(){
    return "You can't divide a number by zero";
  }
}
```

```
import ExceptionHandlingDemo.DivisionException;
import ExceptionHandlingDemo.Calculator;
import java.util.*;
import java.lang.*;

public class CalculatorMain{
  public static void main(String[] ar){
    Scanner scan = new Scanner(System.in);
    Calculator d = new Calculator();

    try{
    System.out.println("Enter two numbers: ");
    int num1 = scan.nextInt();
    int num2 = scan.nextInt();

    if(num2 == 0){
     throw new DivisionException();
     }
    else{
     d.divide(num1, num2);
     }
    }
  catch(DivisionException e){
  System.out.println(e);
  }
}
}
```

Enter two numbers:
12
0
You can't divide a number by zero