1. Write an application that accepts two numbers, divides the first number with the second number throws Arithmetic Exception when divides by zero.

package assign14;

import java.util.Scanner;

public class number {

public static void main(String[] args)

{

int n1,n2;

Scanner sc=new Scanner(System.in);

try {

System.out.println("enter the first number:");

n1=sc.nextInt();

System.out.println("enter the second number:");

n2=sc.nextInt();

int n3=n1/n2;

System.out.println("answer:" +n3);

} catch (ArithmeticException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Divide by zero is an runtime error");

}

}

Output:

enter the first number:

23

enter the second number:

0

java.lang.ArithmeticException: / by zero

Divide by zero is an runtime error

at assign14.number.main(number.java:14)

1. Carrying forward with the above problem, handled ArithmeticException by raising UnsupportedOperationException as a solution.

package assign14;

import java.util.Scanner;

public class number {

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

try {

System.out.println("enter the first number:");

int n1=sc.nextInt();

System.out.println("enter the second number:");

int n2=sc.nextInt();

int n3=n1/n2;

System.out.println("answer:" +n3);

} catch (ArithmeticException e) {

// TODO Auto-generated catch block

e.printStackTrace();

System.out.println("exception handled" +e);

System.out.println("UnsupportedOperationException");

}

}

}

Output:

enter the first number:

12

enter the second number:

0

java.lang.ArithmeticException: / by zero

exception handledjava.lang.ArithmeticException: / by zero

UnsupportedOperationException

at assign14.number.main(number.java:14)

1. Perform withdraw functionality with saving account object.

i)Raise InsufficientBalanceException if you are trying to withdraw more than balance.

**package** number2;

**import** java.util.Scanner;

**class** InsufficientBalanceException **extends** RuntimeException{

}

**public** **class** bank {

Scanner sc = **new** Scanner(System.***in***);

**public** **void** withdrawal(**double** a)

{

System.***out***.println("Enter your Id ");

**long** id = sc.nextLong();

System.***out***.println("Enter your balance");

**double** b = sc.nextDouble();

**try** {

**if**(a<=b) {

b = b - a;

System.***out***.println("Balance= " + b);

}

**else** {

**throw** **new** InsufficientBalanceException();

}

}

**catch** (InsufficientBalanceException e) {

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

bank s = **new** bank();

s.withdrawal(2000);

}

}

Output:

Enter your Id

8

Enter your balance

2500

Balance= 500.0

Enter your Id

7

Enter your balance

500

number2.InsufficientBalanceException

at number2.bank.withdrawal(bank.java:21)

at number2.bank.main(bank.java:32)

ii) Raise a illigalBankTransaction if you are trying to withdraw negative amount from account.

package number3;

import java.util.Scanner;

class IllegalBankTransactionException extends RuntimeException{

}

public class bank {

Scanner sc = new Scanner(System.in);

public void withdrawal(double a)

{

System.out.println("Enter your Id ");

long id = sc.nextLong();

System.out.println("Enter your balance ");

double b = sc.nextDouble();

try {

if(a>0) {

System.out.println("Balance= " + b);

}

else {

throw new IllegalBankTransactionException();

}

}

catch (IllegalBankTransactionException e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

bank s = new bank();

s.withdrawal(-10201);

}}

Output:

Enter your Id

8

Enter your balance

600

number3.IllegalBankTransactionException

at number3.bank.withdrawal(bank.java:19)

at number3.bank.main(bank.java:30)