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

import java.util.Scanner;

public class exce {

public static void main(String[] args)

{

int a,b;

Scanner sc=new Scanner(System.in);

try {

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

a=sc.nextInt();

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

b=sc.nextInt();

int c=a/b;

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

} catch (ArithmeticException e) {

e.printStackTrace();

}

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

}

}

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

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 a=sc.nextInt();

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

int b=sc.nextInt();

int c=a/b;

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

} catch (ArithmeticException e) {

e.printStackTrace();

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

System.out.println("UnsupportedOperationException");

}

}

}

1. Perform withdraw functionality with saving account object.
   1. Raise InsufficientBalanceException if you are trying to withdraw more than balance.

**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) {

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

s.withdrawal(2000);

}

}

* 1. Raise a illigalBankTransaction if you are trying to withdraw negative amount from account.

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);

}}