EXCEPTION HANDLING ASSIGNMENT

1. Write code which result Arithmetic Exception.

ANS:

package pactical;

import java.util.Scanner;

public class Arithmetics {

public static void main(String args[]) {

try{

Scanner sc = new Scanner(System.in);

System.out.println("enter a num");

int num1 = sc.nextInt();

System.out.println("enter a num");

int num2 = sc.nextInt();

int num3 = num1/num2;

System.out.println(num3);

}

}catch(Exception e){

System.out.println(e);

}

}

}

OUTPUT: enter a num

10

enter a num

0

Exception in thread "main" java.lang.ArithmeticException: / by zero

at pactical.Arithmetics.main(Arithmetics.java:17)

1. Carrying forward with the above program handle arithmetic exception with unsupportedOperationalException in solution.

ANS:

package pactical;

import java.util.Scanner;

public class Arithmetics {

public static void main(String args[]) {

try {

Scanner sc = new Scanner(System.in);

System.out.println("enter a num");

int num1 = sc.nextInt();

System.out.println("enter a num");

int num2 = sc.nextInt();

int num3 = num1/num2;

System.out.println(num3);

}

catch(Exception e) {

System.out.println(e);

System.out.println("UnsupportedOperationException") } } }

OUTPUT : enter a num

10

enter a num

0

java.lang.ArithmeticException: / by zero

UnsupportedOperationException

1. Perform withdraw functionality with saving account object .
2. Raise InsufficientBalanceException if you are trying to windraw more than balance.
3. Raise a IlligalBankTransaction if you are trying to withdraw negative amount from account

ANS:

i.

package pactical;

import java.util.Scanner;

class InsufficientBalanceException extends RuntimeException {

}

public class SavingAccount {

Scanner sc = new Scanner(System.in);

public void make\_a\_Withdrawal(double amount) {

System.out.println("enter your id");

long id = sc.nextLong();

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

double balance = sc.nextDouble();

try {

if (amount <= balance) {

balance = balance - amount;

System.out.println("balance= " + balance);

} else {

throw new InsufficientBalanceException();

}

} catch (InsufficientBalanceException e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

SavingAccount sa = new SavingAccount();

sa.make\_a\_Withdrawal(1000);

}

}

**OUTPUT**:

enter your id

12345

enter your balance

100

pactical.InsufficientBalanceException

at pactical.SavingAccount.make\_a\_Withdrawal(SavingAccount.java:29)

at pactical.SavingAccount.main(SavingAccount.java:49)

ii.

package pactical;

import java.util.Scanner;

class IllegalBankTransactionException extends RuntimeException {

}

public class SavingAccount {

Scanner sc = new Scanner(System.in);

public void make\_a\_Withdrawal(double amount) {

System.out.println("enter your id");

long id = sc.nextLong();

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

double balance = sc.nextDouble();

try {

if (amount < 0) {

System.out.println("balance= " + balance);

} else {

throw new IllegalBankTransactionException();

}

} catch (IllegalBankTransactionException e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

SavingAccount sa = new SavingAccount();

sa.make\_a\_Withdrawal(-1012345);

}

}

OUTPUT:

enter your id

1234567

enter your balance

100

pactical.IllegalBankTransactionException

at pactical.SavingAccount.make\_a\_Withdrawal(SavingAccount.java:26)

at pactical.SavingAccount.main(SavingAccount.java:42)