**Assignments on Exception Handling**

1. Write an application that accepts two numbers, divide the first number with the second number and display the result. Hint: You need to handle ArithmeticException which is thrown when there is an attempt to divide a number by a zero.

**package** exception\_handling;

**import** java.util.Scanner;

**public** **class** ArithExcep {

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

**try** {

**int** x,y;

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

System.***out***.println("Enter the first number ");

x = sc.nextInt();

System.***out***.println("Enter the second number ");

y = sc.nextInt();

System.***out***.println(x/y);

}

**catch**(ArithmeticException ex)

{

ex.printStackTrace();

System.***out***.println("Exception Handled When Divion By Zero "+ ex);

}

}

}

OUTPUT:

Enter the first number

7

Enter the second number

0

java.lang.ArithmeticException: / by zero

Exception Handled When Divion By Zero java.lang.ArithmeticException: / by zero

at exception\_handling.ArithExcep.main(ArithExcep.java:18)

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

**package** exception\_handling;

**import** java.util.Scanner;

**public** **class** ArithExcep {

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

**try** {

**int** x,y;

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

System.***out***.println("Enter the first number ");

x = sc.nextInt();

System.***out***.println("Enter the second number ");

y = sc.nextInt();

System.***out***.println(x/y);

}

**catch**(ArithmeticException ex)

{

System.***out***.println("Divided by zero operation can not possible "+ ex);

System.***out***.println("UnsupportedOperationException "+ ex);

}

}

}

OUTPUT:

Enter the first number

13

Enter the second number

0

Divided by zero operation can not possible java.lang.ArithmeticException: / by zero

UnsupportedOperationException java.lang.ArithmeticException: / by zero

3.Perform withdraw functionality with saving account object.

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

**package** exception\_handling;

**import** java.util.Scanner;

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

}

**public** **class** BankAccount {

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) {

BankAccount s = **new** BankAccount();

s.withdrawal(5000);

}

}

OUTPUT:

Enter your Id

23455

Enter your balance

2000

exception\_handling.InsufficientBalanceException

at exception\_handling.BankAccount.withdrawal(BankAccount.java:22)

at exception\_handling.BankAccount.main(BankAccount.java:33)

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

**package** exception\_handling;

**import** java.util.Scanner;

**class** IllegalBankTransactionException **extends** RuntimeException{

}

**public** **class** SavingAccount {

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

SavingAccount s = **new** SavingAccount();

s.withdrawal(-12345);

}

}

OUTPUT:

Enter your Id

34567

Enter your balance

23456

exception\_handling.IllegalBankTransactionException

at exception\_handling.SavingAccount.withdrawal(SavingAccount.java:22)

at exception\_handling.SavingAccount.main(SavingAccount.java:33)