

# CORE JAVA

## EXCEPTION HANDLING ASSIGNMENT

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

Code: -

```
ExceptionDividebyzero.java
1 package exception;
2 import java.util.*;
3 public class ExceptionDividebyzero
4 {
5     public static void main(String args[])
6     {
7         try
8         {
9             Scanner sc = new Scanner(System.in);
10            System.out.println("Enter number 1:");
11            int dividend=sc.nextInt();
12            System.out.println("Enter number 2:");
13            int divisor=sc.nextInt();
14            int result=dividend/divisor;
15            System.out.println("Result:"+result);
16        }
17        catch (ArithmeticException e)
18        {
19            System.out.println("Arithmetic exception occurred");
20            e.printStackTrace();
21        }
22    }
23 }
24
25
```

Problems • Javadoc • Declaration • Console × • Git Staging  
<terminated> ExceptionDividebyzero [Java Application] C:\Users\MBALKRIS\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot  
Enter number 1:  
9  
Enter number 2:  
0  
Arithmetic exception occurred  
java.lang.ArithmeticException: / by zero  
at exception/exception.ExceptionDividebyzero.main(ExceptionDividebyzero.java:14)

```
ExceptionDividebyzero.java
1 package exception;
2 import java.util.*;
3 public class ExceptionDividebyzero
4 {
5     public static void main(String args[])
6     {
7         try
8         {
9             Scanner sc = new Scanner(System.in);
10            System.out.println("Enter number 1:");
11            int dividend=sc.nextInt();
12            System.out.println("Enter number 2:");
13            int divisor=sc.nextInt();
14            int result=dividend/divisor;
15            System.out.println("Result:"+result);
16        }
17        catch (ArithmeticException e)
18        {
19            System.out.println("Arithmetic exception occurred");
20            e.printStackTrace();
21        }
22    }
23 }
24
25
```

Problems • Javadoc • Declaration • Console × • Git Staging  
<terminated> ExceptionDividebyzero [Java Application] C:\Users\MBALKRIS\p  
Enter number 1:  
10  
Enter number 2:  
5  
Result:2

2. Carrying forward with the above problem, handle ArithmeticException by raising UnsupportedOperationException as a solution.

Code: -

```
UnsupportedDividebyzero.java
1 package exception;
2 import java.util.*;
3 public class UnsupportedDividebyzero
4 {
5     public static void main(String args[])
6     {
7         try
8         {
9             Scanner sc = new Scanner(System.in);
10            System.out.println("Enter number 1:");
11            int dividend=sc.nextInt();
12            System.out.println("Enter number 2:");
13            int divisor=sc.nextInt();
14            int result=dividend/divisor;
15            System.out.println("Result:"+result);
16        }
17        catch (UnsupportedOperationException e)
18        {
19            System.out.println("Unsupported exception occurred");
20            e.printStackTrace();
21        }
22    }
23 }
24
25
```

Problems • Javadoc • Declaration • Console × • Git Staging  
<terminated> UnsupportedDividebyzero [Java Application] C:\Users\MBALKRIS\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot  
Enter number 1:  
2  
Enter number 2:  
0  
Exception in thread "main" java.lang.ArithmeticException: / by zero  
at exception/exception.UnsupportedDividebyzero.main(UnsupportedDividebyzero.java:14)

```
UnsupportedDividebyzero.java
1 package exception;
2 import java.util.*;
3 public class UnsupportedDividebyzero
4 {
5     public static void main(String args[])
6     {
7         try
8         {
9             Scanner sc = new Scanner(System.in);
10            System.out.println("Enter number 1:");
11            int dividend=sc.nextInt();
12            System.out.println("Enter number 2:");
13            int divisor=sc.nextInt();
14            int result=dividend/divisor;
15            System.out.println("Result:"+result);
16        }
17        catch (UnsupportedOperationException e)
18        {
19            System.out.println("Unsupported exception occurred");
20            e.printStackTrace();
21        }
22    }
23 }
24
25
```

Problems • Javadoc • Declaration • Console × • Git Staging  
<terminated> UnsupportedDividebyzero [Java Application] C:\Users\MBALKRIS\p  
Enter number 1:  
20  
Enter number 2:  
10  
Result:2

3. Write an application to perform withdraw functionality on a SavingAccount object. Point to note:
  - a. RaiseInsufficientBalanceException if you are trying to withdraw more than balance or when your balance is zero. E.g. if your balance is 2000 and if you are trying to withdraw 2100 or if your balance is zero and you are trying to withdraw positive value.

- b. Raise `IllegalBankTransactionException` if you are trying to withdraw a negative value from your balance. E.g. if you trying to withdraw a negative value `savingAcc.withdraw(-1000)`;

Note: - `SavingAccount`

|--long id

|--double balance

|-- double withdraw (double amount)

|-- double deposit (double amount)

Code: -

`SavingAccount.java`

```
SavingAccount.java x IllegalBankTransactionException.java InSufficientBalanceException.java *Withdraw.java
1 package exception;
2 public class SavingAccount
3 {
4     long loginID= 123;
5     double balance =2000;
6     public void show()
7     {
8         System.out.println("LoginId : " + loginID + "\n" + "Balance : " + balance);
9     }
10    public void withdraw(double amount) throws IllegalBankTransactionException, InSufficientBalanceException
11    {
12        if( balance < amount || balance==0)
13        {
14            throw new InSufficientBalanceException("Insufficient Balance");
15        }
16        else if (amount<0)
17        {
18            throw new IllegalBankTransactionException("Illegal Bank Transaction Exception");
19        }
20        else
21        {
22            balance -= amount;
23            System.out.println("Your money:" + amount);
24        }
25    }
26 }
```

Problems Javadoc Declaration Console x Git Staging

<terminated> Withdraw [Java Application] C:\Users\MBALKRIS\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.0.v202110-1

LoginId : 123  
Balance : 2000.0  
Enter amount to withdraw:  
2500  
Error : Insufficient Balance  
LoginId : 123  
Balance : 2000.0

`IllegalBankTransactionException.java`

`InSufficientBalanceException`

```
SavingAccount.java IllegalBankTransactionException.java x InSufficientBalanceException.java
1 package exception;
2 public class IllegalBankTransactionException extends Exception
3 {
4     public IllegalBankTransactionException(String msg)
5     {
6         super(msg);
7     }
8 }
```

```
SavingAccount.java IllegalBankTransactionException.java InSufficientBalanceException.java x
1 package exception;
2 public class InSufficientBalanceException extends Exception
3 {
4     public InSufficientBalanceException(String msg)
5     {
6         super(msg);
7     }
8 }
```

## Withdraw.java

```

SavingAccount.java      IllegalBankTransactionException.java      InSufficientBalanceException.java      Withdraw.java ×
1 package exception;
2 import java.util.Scanner;
3 public class Withdraw
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8         SavingAccount Account = new SavingAccount();
9         Account.show();
10        System.out.println("Enter amount to withdraw:");
11        double amount = sc.nextDouble();
12        try
13        {
14            Account.withDraw(amount);
15        }
16        catch (InsufficientBalanceException e)
17        {
18            System.out.println("Error : " + e.getMessage());
19        }
20        catch (IllegalBankTransactionException e)
21        {
22            System.out.println("Error : " + e.getMessage());
23        }
24        Account.show();
25    }
26 }

```

Problems Javadoc Declaration Console × Git Staging

```

<terminated> Withdraw [Java Application] C:\Users\MBALKRIS\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32_x86_64\jre\bin\java.exe
LoginId : 123
Balance : 2000.0
Enter amount to withdraw:
-1000
Error : Illegal Bank Transaction Exception
LoginId : 123
Balance : 2000.0

```

Actual output: -

Problems @ Javadoc Declaration Console × Git Staging

```
<terminated> Withdraw [Java Application] C:\Users\MBALKRIS\p2\po  
LoginId : 123  
Balance : 2000.0  
Enter amount to withdraw:  
1000  
Your money:1000.0  
LoginId : 123  
Balance : 1000.0
```