**Day 4 Solutions**

**Solution 50:**

**package** com.hsbc.Day4Solutions;

**import** java.util.Scanner;

/\*

\* Lab Solution 50

\* This code is to understand the concept of Exception Handling in Java

\*/

**public** **class** LabSolution50 {

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

**int** num1;

**int** num2;

//int res=0;

**char** ch;

Calc c = **new** Calc();

**try** (Scanner sc = **new** Scanner(System.***in***)) {

System.***out***.println("Enter Num 1 :");

num1 = sc.nextInt();

System.***out***.println("Enter Num 2 :");

num2 = sc.nextInt();

System.***out***.println("Enter operation :");

ch = sc.next().charAt(0);

}

**switch**(ch) {

**case** '+':

c.add(num1, num2);

**break**;

**case** '/':

c.divide(num1,num2);

**break**;

**default**:

System.***out***.println("Wrong Operation Entered !!");

System.***out***.println("Enter Valid Operation ");

}

}

}

**class** Calc {

**int** a,b,c=0;

**void** add(**int** a, **int** b) {

c = a+b;

System.***out***.println("Addition :"+c);

}

**void** divide(**int** a, **int** b) {

**try**{

c = a/b;

}

**catch**(ArithmeticException e) {

System.***out***.println("Arithmetic Exception Occured ");

}

**finally** {

System.***out***.println("Division :"+c);

}

}

}

**Solution 51:**

**Solution 52:**

**Solution 54:**

**package** com.hsbc.Day4Solutions

**import** java.util.Scanner;

/\*

\* Lab Solution 54

\*/

**public** **class** LabSolution54 {

/\*\*

\* **@param** args

\*/

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

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

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

**boolean** checkb=**true**;

**int** check=1;

**while**(checkb) {

**int** n=scan.nextInt();

**try** {

**int** array[]=**new** **int**[n];

**for**(**int** i=0;i<array.length;i++)

array[i]=scan.nextInt();

System.***out***.println("Enter a location to access value.");

**int** m=scan.nextInt();

System.***out***.println(array[m-1]);

String str = "RUPALI";

System.***out***.println("Enter a location value in String.");

**int** k=scan.nextInt();

System.***out***.println("Length of the String is: " + str.length());

System.***out***.println("The substring is: " + str.substring(k));

System.***out***.println("Enter a divisor.");

**int** divisor=scan.nextInt();

**int** s=5/divisor;

array=**null**;

System.***out***.println(array[9]);

/\*int s=5/0;

int array[]=new int[n];

//for(int i=0;i<array.length;i++)

//array[i]=scan.nextInt();

//int m=array[n+1];

/\*String str = "RUPALI";

System.out.println("Length of the String is: " + str.length());

System.out.println("Length of the substring is: " + str.substring(7));\*/

// int s=5/0;

//array=null;\*/

}

**catch**(ArithmeticException ex){

System.***out***.println("Division by 0 not possible, enter non zero value for divisor");

ex.printStackTrace();

System.***out***.println("wanna see more type of exceptions? Press 1, else 0");

check=scan.nextInt();

**if**(check==0)

checkb=**false**;

**else**

checkb=**true**;

}

**catch**(NegativeArraySizeException ex){

System.***out***.println("Can't declare an array of negative size");

ex.printStackTrace();

System.***out***.println("wanna see more type of exceptions? Press 1, else 0");

check=scan.nextInt();

**if**(check==0)

checkb=**false**;

**else**

checkb=**true**;

}

**catch**(ArrayIndexOutOfBoundsException ex){

System.***out***.println("You are trying to access a location that is not initialized/allocated.");

ex.printStackTrace();

System.***out***.println("wanna see more type of exceptions? Press 1, else 0");

check=scan.nextInt();

**if**(check==0)

checkb=**false**;

**else**

checkb=**true**;

}

**catch**(StringIndexOutOfBoundsException ex){

System.***out***.println("You are trying to access a location that is not initialized/allocated.");

ex.printStackTrace();

System.***out***.println("wanna see more type of exceptions? Press 1, else 0");

check=scan.nextInt();

**if**(check==0)

checkb=**false**;

**else**

checkb=**true**;

}

**catch**(IndexOutOfBoundsException ex){

System.***out***.println("You are trying to access a location that is not initialized/allocated.");

ex.printStackTrace();

System.***out***.println("wanna see more type of exceptions? Press 1, else 0");

check=scan.nextInt();

**if**(check==0)

checkb=**false**;

**else**

checkb=**true**;

}

**catch**(NullPointerException ex){

System.***out***.println("We assigned the array to null because we are proceeding towards the end of program!.");

ex.printStackTrace();

System.***out***.println("wanna see these type of exceptions again? Press 1, else 0");

check=scan.nextInt();

**if**(check==0)

checkb=**false**;

**else**

checkb=**true**;

}

}

}

}

**package** com.hsbc.Day4Solutions;

**import** java.util.Scanner;

/\*

\* Lab Solution 54

\* This code is to understand Various Exceptions in Java

\*/

**public** **class** LabSolution54 {

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

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

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

**try**{

**int** num1=30;

System.***out***.println("Enter 2nd num");

**int** num2=sc.nextInt();

**int** output=num1/num2;

System.***out***.println ("Result: "+output);

**int** a[]=**new** **int**[]{1,2,3,4,5};

//Array has only 5 elements

System.***out***.println("Enter the index to be accessed and array size is 5");

**int** index=sc.nextInt();

System.***out***.println("Element is: "+a[index]);

System.***out***.println("Enter the size of array which is not negative other will throw exception");

**int** arrSize=sc.nextInt();

**int**[] myArray = **new** **int**[arrSize];

String str="beginnersbook";

System.***out***.println(str.length());;

**char** c = str.charAt(0);

System.***out***.println("Enter the index to be accessed from string");

**int** index2=sc.nextInt();

c = str.charAt(index2);

System.***out***.println("character is : "+c);

String str2 = "Java Code Geeks";

System.***out***.println("Enter the index to get the substring");

**int** index3=sc.nextInt();

//The following statement throws an exception, because the request index is invalid.

String subStr = str2.substring(index3);

System.***out***.println("Substring is: "+subStr);

}

**catch**(ArithmeticException e){

System.***out***.println ("You Shouldn't divide a number by zero");

e.printStackTrace();

}

**catch**(ArrayIndexOutOfBoundsException e){

System.***out***.println ("ArrayIndexOutOfBounds");

e.printStackTrace();

}

**catch**(StringIndexOutOfBoundsException e){

System.***out***.println("StringIndexOutOfBoundsException!!");

e.printStackTrace();

}

**catch**(NullPointerException e){

System.***out***.println("NullPointerException..");

e.printStackTrace();

}

**catch** (NegativeArraySizeException ex) {

System.***out***.println("Can't create array of negative size");

ex.printStackTrace();

}

}

}

**Solution 56:**

**package** com.hsbc.Day4Solutions;

/\*

\* Lab Solution 56

\* This code is to understand concept of finally keyword

\*/

**public** **class** LabSolution56 {

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

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

**new** DemoFinally(7);

}

}

**class** DemoFinally{

String result;

**int** intArray[]= {1,2,3,4},j,m;

**public** DemoFinally( **int** j) {

**super**();

**this**.j = j;

**try** {

m=10/j;

result=result + j;

System.***out***.println(result);

}

**catch**(Exception e){

j=0;

System.***out***.println("Divisor Should not be 0.Returning exit code" + j );

}

**finally** {

**for**(**int** i=0;i<intArray.length;i++)

System.***out***.println(intArray[i]);

result=**null**;

intArray=**null**;

System.***out***.println(result);

System.***out***.println(intArray);

System.***out***.println("We can see that the object values are now null after its usage!!!");

}

}

}

**Solution 57:**

**package** com.hsbc.Day4Solutions;

/\*

\* Lab Solution 57

\* This code is to understand concept of finally keyword using toString method

\*/

**public** **class** LabSolution57 {

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

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

FinallyKeyword t1=**new** FinallyKeyword();

t1.customMethod(2);

FinallyKeyword t2=**new** FinallyKeyword();

t2.customMethod(0);

}

}

**class** FinallyKeyword{

**int** divisor,j=0;

String result="The result is ",s;

@SuppressWarnings("finally")

**public** String customMethod(**int** divisor) {

//super();

**this**.divisor = divisor;

**try** {

j=10/divisor;

result=result + j;

System.***out***.println(result);

}

**catch**(Exception e){

j=0;

System.***out***.println("Divisor should not be Zero. Returning exit code" + j);

}

**finally** {

s="Execution is finished";

**return** s;

}

}

}