|  |  |  |
| --- | --- | --- |
| **1305104 Object-Oriented Design and Programming Laboratory** |  | |
| **Laboratory 11:** Exception | **School of Information Technology** | |
| **Name:** | **ID:** | **Section:** |
| **Date:** | **Due date:** | |

**Exception**

Exception is the problem found during the execution of the program. It is the problem that can be handled and then the program can continue.

Example 1

Try the program below.

|  |
| --- |
| public class MyException {  public void run(){  System.out.println(5/0);  }  } |

|  |
| --- |
| public class Main  {  public static void main(String[] args){  MyException me = new MyException();  me.run();  }  } |

Question

1. What is the problem after running class Main? What type of exception is found?

|  |
| --- |
|  |

Then try to handle this exception by changing codes of class MyException as follow.

|  |
| --- |
| public class myException {  public void run(){  try {  System.out.println(5/0);  }  catch(ArithmeticException e){  System.out.println("Cannot divide by zero");  }  }  } |

1. What is the new result?

|  |
| --- |
|  |

Example 2

Try the program below.

|  |
| --- |
| import java.util.Scanner;  public class InputException {  public void run(){  Scanner sc = new Scanner(System.in);  System.out.print("Enter an integer...");  int num = sc.nextInt();  System.out.println("You input "+num);  }  } |

|  |
| --- |
| public class Main {  public static void main(String[] args){  InputException ie = new InputException();  ie.run();  }  } |

Question

1. What is the problem after running class Main and enter “a”? What type of exception is found?

|  |
| --- |
|  |

Then try to handle this exception by changing codes of class MyException as follow.

|  |
| --- |
| import java.util.Scanner;  import java.util.InputMismatchException;  public class InputException {  public void run(){  Scanner sc = new Scanner(System.in);  boolean goodInput = false;  do {  try {  System.out.print("Enter an integer...");  int num = sc.nextInt();  System.out.println("You input "+num);  goodInput = true;  }  catch(InputMismatchException e){  sc.nextLine(); //to clear \n from scanner  System.out.println("Input is not a number");  System.out.println("Try again");  }  }while(goodInput==false);  }  } |

1. Now, what is the problem after running class Main and enter “a”?

|  |
| --- |
|  |

Example 3

This example shows how to throw exception in computing factorial. Here we cannot compute factorial of negative number. So we throw the exception for it.

|  |
| --- |
| public class Factorial  {  public void computeFactorial(int num) throws Exception {  if(num<0){  System.out.println("Cannot find factorial of negative numbers");  throw new Exception();  }  Else {  int fac = 1;  for(int i=1;i<=num;i++)  fac \*= i;  System.out.println("Factorial of "+num+" = "+fac);  }  }  } |

|  |
| --- |
| public class Main {  public static void main(String[] args){  Factorial ft = new Factorial();  Try {  ft.computeFactorial(5);  ft.computeFactorial(-5);  }  catch(Exception e){  System.out.println("Computation Error");  }  }  } |

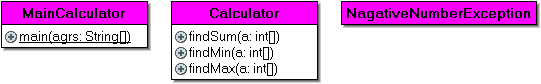
Question

1. What is the result?

|  |
| --- |
|  |

Question 11.1

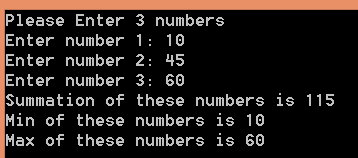
Create a calculator program following the class diagram below.

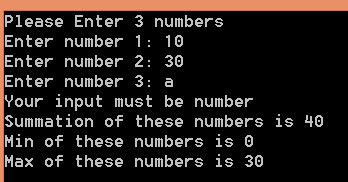


The program must include

* Array of 3 input numbers (enter from keyboard with using loop)
* InputMismatchException and Your own exception.
* throws/throw/ try+catch

Expected output:





Put the complete code :

//import statements

class NagativeNumberException extends Exception{

}

class Calculator{

public void findSum(int a[]) throws NagativeNumberException{

// add code here

}

public void findMin(int a[]) throws NagativeNumberException{

// add code here

}

public void findMax(int a[]) throws NagativeNumberException{

//add code here

}

public class MainCalculator {

public static void main(String[] agrs){

Calculator cal = new Calculator();

Scanner sc = new Scanner(System.in);

int data[] = new int[\_\_\_\_];

System.out.println("Please Enter 3 numbers");

for(int i=0; i<data.length; i++){

System.out.print("Enter number "+\_\_\_\_\_\_\_\_+": ");

try{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}catch(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_){

System.out.println("Your input must be number ");

}

}

try{

cal.findSum(data);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}catch(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_){

System.out.println("Numbers cannot be nagative ");

}

}

}

Question 11.2

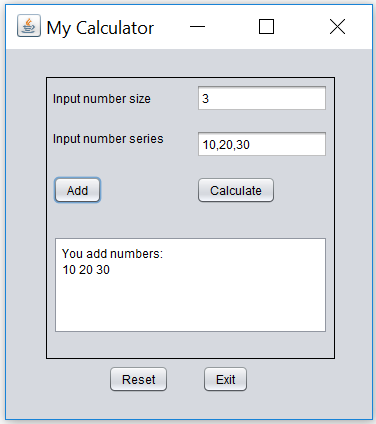
Create a calculatorGUI program that performs as shown in the figures below.

Hint:

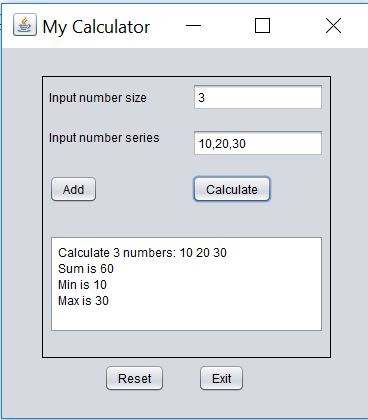
- Use findMax(),findMin(),and FindSum() from the previous question and add them to class “CalculatorGUI” (save as CalculatorGUI.java)

- Use keyword “string to array of int java” to search the appropriate java method() to convert string to array of integer numbers.

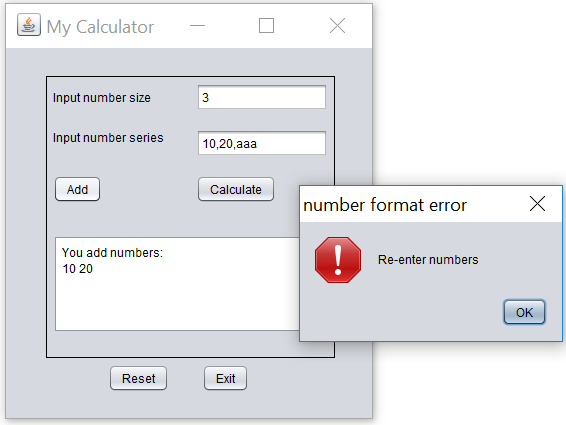
- Apply the appropriate Exception when you input string rather than integer.



Click **Add** then show these input numbers



Click **Calculate** then show sum,min,and max results



Input a string, click **Add**, and then show error in the message dialog as shown

**CalculatorGUI.java**

public class CalculatorGUI{

public int findSum(int a[]) {

}

public int findMin(int a[]){

}

public int findMax(int a[]){

}

}

JFrame code parts:

Put import statements and variable declaration (top of the program)

Put “Add” button code

Put “Calculate” button code

Show the result here:

**Sending Method:**

* Save this document file that contains you all answers as “**OODP\_Lab11\_yourID.docx and Lab11\_YourID.jar**” (Do not zip the files)
* And upload it to the google classroom.