# **Exceptions**

1. Write a program that repeatedly asked for the numerator and divisor. For each set of data, the program prints out the result (quotient), or an informative error message if there is a problem (division by zero or poor input data). The program continues looping, even if there is a problem. Exit the loop when data entered for the numerator start with characters “q” or “Q”. Don’t print out an error message in this case. Don’t ask for the divisor if the user just asked to quit.

Here is sample output from one run:

Enter the numerator: 12

Enter the divisor: 4

12 / 4 is 3

Enter the numerator: 12

Enter the divisor: 0

You can't divide 12 by 0

Enter the numerator: glarch

You entered bad data.

Please try again.

Enter the numerator: quit

| /\*  \* Program name: DivisiorProgram.java  \*  \* By: Lucas Chow (Last edited: 2022-09-30)  \*  \* ICS4U1 - 04\_Gr11Review  \*  \* This program repeatedly prompts the user for a numerator and divisor and prints out the quotient  \* If bad data is entered, for example 0, (1/0 is undefined), it tells the user to try again,  \* not crashing the program. If the user enters 'quit' the program  \*  \*  \*/  //importing the scanner  import java.util.Scanner;  public class DivisiorProgram  {  /\*  \* boolean isInt(String input)  \*  \* returns true or false if string can be parsed to int  \*  \* String input - the input string to check if it can be parsed  \*  \* This method takes in the parameter String input, and checks if it can be parsed  \* to an integer.  \*  \* \*/  public static boolean isInt(String input)  {  boolean isIntValid = false;  try {  Integer.parseInt(input);  isIntValid = true;  } catch (NumberFormatException e)  {    }  return isIntValid;  }    public static void main(String[] args)  {  //initializing variables and objects  Scanner sc = new Scanner(System.in);  boolean programNotQuit = true;  String userInput;  int numerator;  int denominator;  double quotient;    //This is required for some reason so the program doesn't crash (for conditional loops)  numerator = 0;    do  {  System.out.print("\nEnter the numerator: ");  userInput = sc.nextLine();  numerator = 0;  //Setting the numerator to the input if the input is valid.  if (( (int)userInput.charAt(0) == 113 || (int)userInput.charAt(0) == 81) && userInput.length() == 1)  {  programNotQuit = false;  }    //makes sure numerator is valid  while (!isInt(userInput) && programNotQuit)  {  System.out.println("You entered bad data\nPlease try again");  System.out.print("\nEnter the numerator: ");  userInput = sc.nextLine();  }    //if valid, parses  if (isInt(userInput))  {  numerator = Integer.parseInt(userInput);  }    denominator = 0;    //while program is running properly  if (programNotQuit)  {  System.out.print("Enter the denominator: ");  userInput = sc.nextLine();  if (((int)userInput.charAt(0) == 113 || (int)userInput.charAt(0) == 81) && userInput.length() == 1)  {  programNotQuit = false;  }  while (!isInt(userInput) && programNotQuit)  {  System.out.println("You entered bad data\nPlease try again");  System.out.print("\nEnter the denominator: ");  userInput = sc.nextLine();  if (((int)userInput.charAt(0) == 113 || (int)userInput.charAt(0) == 81) && userInput.length() == 1)  {  programNotQuit = false;  break;  }  }    //checks for denominator  if (isInt(userInput) && programNotQuit)  {  denominator = Integer.parseInt(userInput);  if (!isInt(userInput))  {  programNotQuit = false;    System.out.println("You entered bad data\nPlease try again");  }  else if (isInt(userInput) && Integer.parseInt(userInput)\*1 == 0)  {  programNotQuit = false;  System.out.println("You can't divide "+numerator+" by 0");  }    //makes sure denomiator is interger AND non-zer onumber  while (!programNotQuit)  {  System.out.print("\nEnter the denominator: ");  userInput = sc.nextLine();  if (isInt(userInput) && Integer.parseInt(userInput) == 0)  {  System.out.println("demoniator can't be zero!!!!!!!");  }  else if (!isInt(userInput))  {  System.out.println("NO BAD DATA OR ELSE !!! >:(");  }  else if (isInt(userInput) && Integer.parseInt(userInput) != 0)  {  System.out.println("FINALLY CORRECT DATA!!!!");  denominator = Integer.parseInt(userInput);  programNotQuit = true;  }  }  }    }  if (programNotQuit)  {    //calculating quotient  quotient = (double)numerator / (double)denominator;  System.out.printf("The value of %d / %d is %.2f\n",numerator, denominator, quotient);  }    } while (programNotQuit);    //ends program if input is q  System.out.println("Ending program!");  sc.close();  }  } |
| --- |
|  |

1. Write a program that reads from a file “input.txt” a list of numbers that are arranged into groups of various sizes. The program outputs the sum of the numbers in each group. Each group starts with a one-line descriptive phrase. The phrase can be anything that is not a number. Some groups may have zero number in them.

input.txt:

Group A

23

-12

29

-84

Group B

-2

-45

-90

123

26

19

-5

-30

9

Group C

Last Group

12

-34

23

47

52

8

Corresponding Output:

Group A

Sum = -44

Group B

Sum = 5

Group C

Sum = 0

Last Group

Sum = 108

| /\*  \* Program name: TotalOfGroup.java  \*  \* By: Lucas Chow (Last edited: 2022-09-30)  \*  \* ICS4U1 - 04\_Gr11Review  \*  \* Reads the file of input.txt, and for each group and their undifined  \* preceding numbers, calulates the sum of those preceeding numbers with their respective groupname  \*  \*/  //importing BufferedReader  import java.io.\*;  public class TotalOfGroup  {  /\*  \* boolean isInt(String input)  \*  \* returns true or false if string can be parsed to int  \*  \* String input - the input string to check if it can be parsed  \*  \* This method takes in the parameter String input, and checks if it can be parsed  \* to an integer.  \*  \* \*/    public static boolean isInt(String x)  {  boolean trueOrNot = false;  try  {  Integer.parseInt(x);  trueOrNot= true;  } catch (NumberFormatException e) {}  return trueOrNot;  }      public static void main(String[] args)  {  //initializing variables and objects  String fileName = "input.txt";  int sum;  String input;  String groupName;  try  {    //creating buffered reader  BufferedReader in = new BufferedReader(new FileReader(fileName));  input = in.readLine();    //reading the lines  while (input != null)  {  if (!isInt(input))  {  //assigning groupname to input  groupName = input;  sum = 0;  input = in.readLine();  //while input is a integer, adds it to sum  while (isInt(input))  {  sum += Integer.parseInt(input);  input = in.readLine();  }    //prints out groupname and sum  System.out.println(groupName);  System.out.println("Sum = "+sum+"\n");  }  }      in.close();  } catch (IOException e)  {  System.out.println(e+ " error reading file " + fileName);  }  }  } |
| --- |
|  |