

Problem 1

```
package problem1;

import java.util.Scanner;

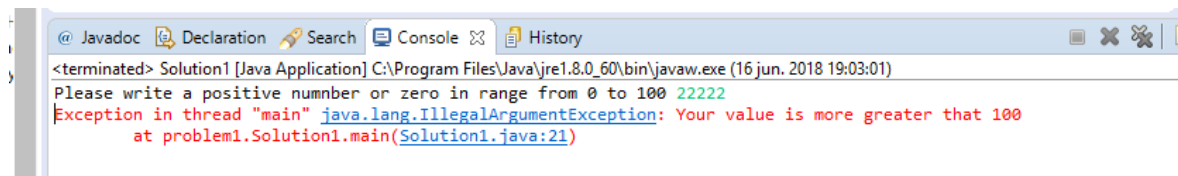
public class Solution1 {
    public static void main(String[] args) {
        System.out.print("Please write a positive numnber or zero in range from 0 to 100 ");
        Scanner sc = new Scanner(System.in);

        String ch = sc.next();

        if(!ch.matches("[\\-d|\\d]+$"))
        {
            throw new IllegalArgumentException("Only allow numeric values");
        }

        Integer value=Integer.parseInt(ch);
        if(value>100)
        {
            throw new IllegalArgumentException("Your value is more greater that 100 ");
        }

        if(value<0)
        {
            throw new IllegalArgumentException(" Upps your value is les that zero");
        }
        else
        {
            System.out.printf("Upa your value is correct %s \\n" ,value );
            return ;
        }
    }
}
```



The screenshot shows a Java IDE window with a console tab. The console output displays the program's execution, including a prompt for input, the user's input '22222', and a runtime exception: 'java.lang.IllegalArgumentException: Your value is more greater that 100'. The exception message is highlighted in red, and the stack trace points to the file 'Solution1.java' at line 21.

```
@ Javadoc Declaration Search Console History
<terminated> Solution1 [Java Application] C:\Program Files\Java\jre1.8.0_60\bin\javaw.exe (16 jun. 2018 19:03:01)
Please write a positive numnber or zero in range from 0 to 100 22222
Exception in thread "main" java.lang.IllegalArgumentException: Your value is more greater that 100
    at problem1.Solution1.main(Solution1.java:21)
```

Problem 2

```
package problem2;

import java.util.Scanner;

public class CustomerAccount {
    String Cus_name ;
    String Acc_No ;
    Double Balance ;
    private final Double MAX_VALUE_ON_ACCCOUNT=100.0;

    public CustomerAccount(String cus_name, String acc_No, Double balance) {

        if (balance<this.MAX_VALUE_ON_ACCCOUNT )
        {
            throw new CustomerAccountException(String.format("Withdraw :Balance Initial
reach below 100$ : to new customer %s initial is %s Check and try again ", cus_name,balance));
        }

        this.Cus_name = cus_name;
        this.Acc_No = acc_No;
        this.Balance = balance;
    }
    @Override
    public String toString() {

        return String.format("|Name(%s) :<Id:%s, Balance:$%,.2f>|",
this.Cus_name,this.Acc_No,this.Balance);
    }

    public boolean Withdraw(Double value)
    {

        if (value<=0)
        {
            throw new CustomerAccountException("Withdraw : only positive values are
allowed");
        }

        if (value > this.Balance)
        {
            throw new CustomerAccountException(String.format("Withdraw (Withdraw amount
exceed the balance):Amount not allowed : can not Witdraw %s because your balance is %s",
value,this.Balance) );
        }

        if (this.Balance-value>=this.MAX_VALUE_ON_ACCCOUNT )
        {
            throw new CustomerAccountException(String.format("Withdraw :Balance reach
below 100$ : request is %s and your current balance %s", value,this.Balance));
        }

        this.Balance=this.Balance-value;
        return true;
    }
}
```

```

    }

    public boolean deposit(Double value)

    {

        if (value<=0)
        {
            throw new CustomerAccountException(String.format("Deposit : only positive
values are allowed , value <%s>", value) );
        }

        this.Balance=this.Balance+value;
        return true;
    }

    public static void main(String[] args) {
        CustomerAccount pedro= new CustomerAccount("Pedro", "A32241E", 170.0);
        pedro.deposit(130.0);

        try
        {
            pedro.deposit(320.0);
        }
        catch (CustomerAccountException e)
        {
            System.err.println(e.getMessage()+"\n");
        }

        try
        {
            pedro.deposit(-92.0);
        }
        catch (CustomerAccountException e)
        {
            System.err.println(e.getMessage()+"\n");
        }

        try
        {
            pedro.Withdraw(700.0);
        }
        catch (CustomerAccountException e)
        {
            System.err.println(e.getMessage()+"\n");
        }

        System.out.println(pedro);

        try
        {
            new CustomerAccount("Farid", "A232241E", 10.0);
        }
        catch (CustomerAccountException e)
        {

```

```

        System.err.println(e.getMessage()+"\n");
    }

}

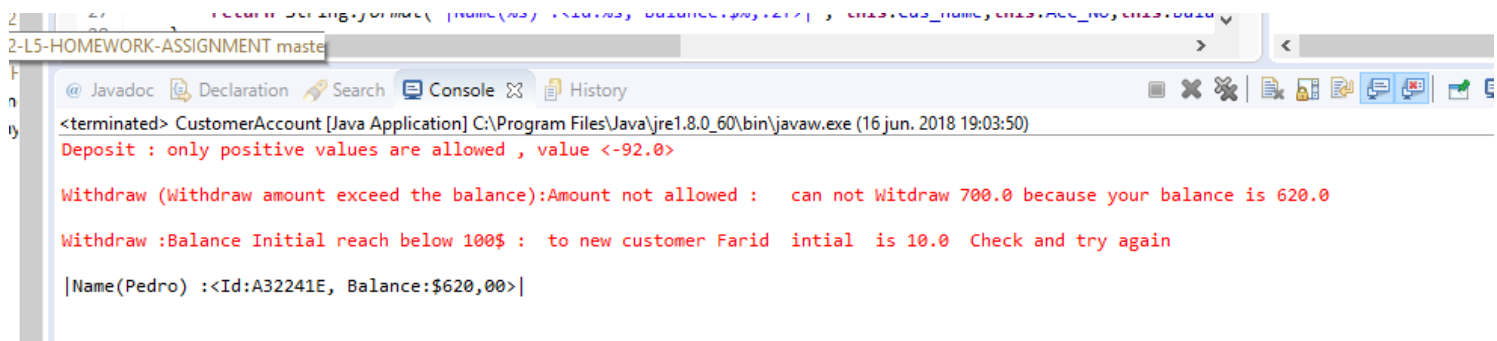
package problem2;

public class CustomerAccountException extends IllegalArgumentException {

    public CustomerAccountException(String s)
    {

        super(s);
    }
}

```



The screenshot shows an IDE console window with the following content:

```

2-L5-HOMEWORK-ASSIGNMENT master
@ Javadoc Declaration Search Console History
<terminated> CustomerAccount [Java Application] C:\Program Files\Java\jre1.8.0_60\bin\javaw.exe (16 jun. 2018 19:03:50)
Deposit : only positive values are allowed , value <-92.0>

Withdraw (Withdraw amount exceed the balance):Amount not allowed :   can not Witdraw 700.0 because your balance is 620.0

Withdraw :Balance Initial reach below 100$ :   to new customer Farid  intial  is 10.0  Check and try again

|Name(Pedro) :<Id:A32241E, Balance:$620,00>|

```