

# Rajalakshmi Engineering College

Name: jayaraman S  
Email: 240701210@rajalakshmi.edu.in  
Roll no: 2116240701210  
Phone: 9080708226  
Branch: REC  
Department: CSE - Section 5  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 8\_Q1

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### **Section 1 : Coding**

##### **1. Problem Statement**

Write a program to validate the email address and display suitable exceptions if there is any mistake.

Create 3 custom exception classes as below

DotExceptionAtTheRateExceptionDomainException

A typical email address should have a ". " character, and a "@" character, and also the domain name should be valid. Valid domain names for practice be 'in', 'com', 'net', or 'biz'.

Display Invalid Dot usage, Invalid @ usage, or Invalid Domain message based on email id.

Get the email address from the user, validate the email by checking the

above-mentioned criteria, and print the validity status of the input email address.

#### ***Input Format***

The first line of input contains the email to be validated.

#### ***Output Format***

The output prints a Valid email address or an Invalid email address along with the suitable exception

If email ends with . or contains not exactly one . after @, it throws:

DotException: Invalid Dot usage

Invalid email address

If @ appears not exactly once, it throws:

AtTheRateException: Invalid @ usage

Invalid email address

If the part after the last dot is not among accepted domains:

DomainException: Invalid Domain

Invalid email address

If all conditions satisfied then print:

Valid email address

Refer to the sample input and output for format specifications.

### **Sample Test Case**

Input: sample@gmail.com

Output: Valid email address

### **Answer**

```
// You are using Java
import java.lang.Throwable;
import java.util.Scanner;

class DotException extends Exception{
    DotException(String msg){super(msg);}
}

class AtTheRateException extends Exception{
    AtTheRateException(String msg){super(msg);}
}

class DomainException extends Exception{
    DomainException(String msg){super(msg);}
}

class ValidityTest{
    String email;
    ValidityTest(String email){this.email = email;}

    void checkValidDot() throws DotException{
        if(email.endsWith(".")) || !email.matches("^[^.]+\\.[^.]+$")){
            throw new DotException("DotException: Invalid Dot usage\nInvalid email address");
        }
    }

    void checkValidAt() throws AtTheRateException{
        if(!email.matches("[^@]+@[^.]+[^@]+$")){
            throw new AtTheRateException("AtTheRateException: Invalid @ usage\nInvalid email address");
        }
    }
}
```

```
    }

    void checkValidDomain() throws DomainException{
        if(!email.endsWith("in")||email.endsWith("com")||email.endsWith("net")||email.endsWith("biz")){
            throw new DomainException("DomainException: Invalid Domain\nInvalid email address");
        }
    }
}

public class Main{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        try{
            String email = sc.nextLine();
            ValidityTest test = new ValidityTest(email);
            test.checkValidDot();
            test.checkValidAt();
            test.checkValidDomain();
            System.out.println("Valid email address");
        }catch (DotException e){ System.out.println(e.getMessage());}
        catch (AtTheRateException e){ System.out.println(e.getMessage());}
        catch (DomainException e){ System.out.println(e.getMessage());}
        sc.close();
    }
}
```

**Status : Correct**

**Marks : 10/10**