

# Rajalakshmi Engineering College

Name: Parameswari P

Email: 240701378@rajalakshmi.edu.in

Roll no: 240701378

Phone: 9500133836

Branch: REC

Department: CSE - Section 1

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

DotException  
AtTheRateException  
DomainException

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**

```
import java.util.Scanner;
class DotException extends Exception {
    public DotException(String message) {
        super(message);
    }
}
class AtTheRateException extends Exception {
    public AtTheRateException(String message) {
        super(message);
    }
}
class DomainException extends Exception {
    public DomainException(String message) {
        super(message);
    }
}
class EmailValidator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String email = scanner.nextLine();
        try {
            validateEmail(email);
            System.out.println("Valid email address");
        } catch (DotException | AtTheRateException | DomainException e) {
            System.out.println(e.getMessage());
            System.out.println("Invalid email address");
        }
    }
    private static void validateEmail(String email) throws DotException,
    AtTheRateException, DomainException {
        if (email.chars().filter(ch -> ch == '@').count() != 1) {
            throw new AtTheRateException("AtTheRateException: Invalid @ usage");
        }
    }
}
```

```
        }
        String[] parts = email.split("@");
        String localPart = parts[0];
        String domainPart = parts[1];
        if (localPart.isEmpty() || domainPart.isEmpty() || email.startsWith(".") || email.endsWith(".") || email.startsWith("@") || email.endsWith("@")) {
            throw new DotException("DotException: Invalid Dot usage");
        }
        if (email.contains(..) || email.contains(@@)) {
            throw new DotException("DotException: Invalid Dot usage");
        }
        if (!domainPart.contains(".")) {
            throw new DotException("DotException: Invalid Dot usage");
        }
        String[] domainParts = domainPart.split("\\.");
        String domainExtension = domainParts[domainParts.length - 1];
        if (domainParts.length < 2 || !isValidDomain(domainExtension)) {
            throw new DomainException("DomainException: Invalid Domain");
        }
        if (domainPart.indexOf('.') != domainPart.lastIndexOf('.')) {
            throw new DotException("DotException: Invalid Dot usage");
        }
    }

private static boolean isValidDomain(String domain) {
    return domain.equals("in") || domain.equals("com") || domain.equals("net") || domain.equals("biz");
}
```

**Status :** Correct

**Marks :** 10/10