

# Rajalakshmi Engineering College

Name: Shane A  
Email: 241501197@rajalakshmi.edu.in  
Roll no: 241501197  
Phone: 8300409580  
Branch: REC  
Department: AI & ML - Section 1  
Batch: 2028  
Degree: B.E - AI & ML

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.*;
import java.util.regex.*;

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

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

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String email = sc.nextLine().trim();
        try {
            validateEmail(email);
            System.out.println("Valid email address");
        } catch (DotException e) {
            System.out.println("DotException: " + e.getMessage());
            System.out.println("Invalid email address");
        } catch (AtTheRateException e) {
```

```

        System.out.println("AtTheRateException: " + e.getMessage());
        System.out.println("Invalid email address");
    } catch (DomainException e) {
        System.out.println("DomainException: " + e.getMessage());
        System.out.println("Invalid email address");
    }
}

```

```

static void validateEmail(String email) throws DotException,
AtTheRateException, DomainException {
    if (email.startsWith(".") || email.endsWith(".") || email.contains("..")) {
        throw new DotException("Invalid Dot usage");
    }

    if (email.startsWith("@") || email.endsWith("@") || email.contains("@@") ||
email.chars().filter(c -> c == '@').count() != 1) {
        throw new AtTheRateException("Invalid @ usage");
    }
}

```

```

String[] validDomains = {"in", "com", "net", "biz"};
String[] parts = email.split("@");
if (parts.length != 2) {
    throw new AtTheRateException("Invalid @ usage");
}

```

```

String domainPart = parts[1];
if (!domainPart.contains(".")) {
    throw new DotException("Invalid Dot usage");
}

```

```

int lastDot = domainPart.lastIndexOf('.');
if (lastDot == -1 || lastDot == domainPart.length() - 1) {
    throw new DotException("Invalid Dot usage");
}

```

```

String domainExt = domainPart.substring(lastDot + 1);
boolean valid = false;
for (String d : validDomains) {
    if (d.equals(domainExt)) {
        valid = true;
        break;
    }
}

```

```
}  
    if (!valid) {  
        throw new DomainException("Invalid Domain");  
    }  
  
    String emailRegex = "^[a-zA-Z0-9_+&*~]+(?:\\.[a-zA-Z0-9_+&*~]+)*@(?:[a-zA-Z0-9-]+\\.[a-zA-Z]{2,7})$";  
    Pattern pattern = Pattern.compile(emailRegex);  
    Matcher matcher = pattern.matcher(email);  
  
    if (!matcher.matches()) {  
        throw new AtTheRateException("Invalid email format");  
    }  
}
```

**Status :** Correct

**Marks :** 10/10