

# Rajalakshmi Engineering College

Name: Roshan S  
Email: 241501170@rajalakshmi.edu.in  
Roll no: 241501170  
Phone: 9677031330  
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.*;

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();
        sc.close();

        try {
            validateEmail(email);
            System.out.println("Valid email address");
        } catch (DotException e) {
            System.out.println("DotException: " + e.getMessage());
            System.out.println("Invalid email address");
        }
    }

    private static void validateEmail(String email) {
        if (email == null || email.isEmpty()) {
            throw new DotException("Email cannot be empty");
        }
        if (!email.matches("^[a-zA-Z0-9_+@-]+@[a-zA-Z0-9-]+\\.?[a-zA-Z0-9-]+\\.?[a-zA-Z]{2,}$")) {
            throw new DotException("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");
    }
}

public static void validateEmail(String email) throws DotException,
AtTheRateException, DomainException {
    // Check @ usage
    int atCount = email.length() - email.replace("@", "").length();
    if (atCount != 1 || email.startsWith("@") || email.endsWith("@") ||
email.contains("@@")) {
        throw new AtTheRateException("Invalid @ usage");
    }

    // Check dot usage
    if (email.endsWith(".") || email.startsWith(".") || email.contains("..")) {
        throw new DotException("Invalid Dot usage");
    }

    int atIndex = email.indexOf('@');
    String afterAt = email.substring(atIndex + 1);

    if (!afterAt.contains(".")) {
        throw new DotException("Invalid Dot usage");
    }

    // Check domain
    int lastDot = email.lastIndexOf('.');
    String domain = email.substring(lastDot + 1);

    List<String> validDomains = Arrays.asList("in", "com", "net", "biz");
    if (!validDomains.contains(domain)) {
        throw new DomainException("Invalid Domain");
    }
}
}

```

**Status : Correct**

**Marks : 10/10**