

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Spring,Year:2025),B.Sc.in CSE (Day)**

**LAB REPORT NO - 03**

**Course Title: Data Communication Lab**

**Course Code:** **CSE307 Section:** **223-D1**

**Lab Experiment Name :**  **Write a java program that prompts the user to enter an IP address belonging to a specific class (A, B, or C) and then offers options to convert it into binary or vice versa**

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | **MD.SHAJALAL** | **223002088** |

**Lab Date : 06 – 03 - 2025**

**Submission Date : 10 – 04 - 2024**

**Course Teacher’s Name : Md.Samin Hossain Utsho**

|  |
| --- |
| **Lab Report Status**  **Marks: ………………………………… Signature:.....................**  **Comments:.............................................. Date:..............................** |

**1.Experiment Name:**

Write a java program that prompts the user to enter an IP address belonging to a specific class (A, B, or C) and then offers options to convert it into binary or vice versa

**2.Objective:**

To develop a Java program that:

* Accepts an IP address as input and identifies its class (A, B, or C).
* Provides options to:
  + Convert a decimal IP address to its binary equivalent.
  + Convert a binary-form IP address to its decimal equivalent.

**3. Introduction:**

An **IP (Internet Protocol) address** is a 32-bit unique identifier assigned to devices on a network. It is conventionally expressed in dotted decimal notation (e.g., 192.168.1.1). IP addresses are divided into classes (A, B, C, D, E), primarily used for network classification and addressing.

In networking, conversion between **binary and decimal** forms of IP addresses is essential for low-level configuration, subnetting, and routing. This lab aims to simulate the recognition and conversion process via a Java-based CLI program.

### **4. Theoretical Background:**

#### 4.1 IP Address Structure

* **Format**: Four octets (8 bits each) separated by dots (e.g., 192.168.1.1)
* **Range**: 0.0.0.0 to 255.255.255.255

#### 4.2 IP Address Classes (Relevant to this lab)

| **Class** | **First Octet Range** | **Default Subnet Mask** | **Use Case** |
| --- | --- | --- | --- |
| A | 1 – 126 | 255.0.0.0 | Large Networks |
| B | 128 – 191 | 255.255.0.0 | Medium Networks |
| C | 192 – 223 | 255.255.255.0 | Small Networks |

#### 4.3 Binary Conversion

Each octet is represented as an 8-bit binary number.  
Example:  
192 → 11000000  
168 → 10101000  
Full IP: 192.168.0.1 → 11000000.10101000.00000000.00000001

**5.Tools and Technologies Used:**

 Programming Language: Java

 IDE: Any Java-supported IDE (e.g., IntelliJ, Eclipse, NetBeans)

 JDK Version: 8 or higher

 Operating System: Windows/Linux/MacOS

**6. Java Program Code:**

import java.util.Scanner;

public class  lab {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("=== IP Address Utility ===");

        System.out.print("Enter a valid IP address (decimal form): ");

        String ipAddress = scanner.nextLine();

        if (!isValidIP(ipAddress)) {

            System.out.println("Invalid IP format!");

            return;

        }

        String ipClass = getIPClass(ipAddress);

        if (ipClass == null) {

            System.out.println("IP does not belong to Class A, B, or C.");

            return;

        }

        System.out.println("Detected IP Class: " + ipClass);

        System.out.println("Select an option:");

        System.out.println("1. Convert IP to Binary");

        System.out.println("2. Convert Binary to Decimal IP");

        int choice = scanner.nextInt();

        scanner.nextLine(); // consume newline

        switch (choice) {

            case 1:

                System.out.println("Binary Format: " + decimalToBinary(ipAddress));

                break;

            case 2:

                System.out.print("Enter Binary IP (e.g. 11000000.10101000.00000001.00000001): ");

                String binaryInput = scanner.nextLine();

                if (isValidBinaryIP(binaryInput)) {

                    System.out.println("Decimal Format: " + binaryToDecimal(binaryInput));

                } else {

                    System.out.println("Invalid Binary IP Format!");

                }

                break;

            default:

                System.out.println("Invalid option!");

        }

        scanner.close();

    }

    static boolean isValidIP(String ip) {

        String[] octets = ip.split("\\.");

        if (octets.length != 4) return false;

        for (String octet : octets) {

            try {

                int val = Integer.parseInt(octet);

                if (val < 0 || val > 255) return false;

            } catch (NumberFormatException e) {

                return false;

            }

        }

        return true;

    }

    static boolean isValidBinaryIP(String binaryIP) {

        String[] parts = binaryIP.split("\\.");

        if (parts.length != 4) return false;

        for (String part : parts) {

            if (part.length() != 8 || !part.matches("[01]{8}")) return false;

        }

        return true;

    }

    static String getIPClass(String ip) {

        int firstOctet = Integer.parseInt(ip.split("\\.")[0]);

        if (firstOctet >= 1 && firstOctet <= 126) return "A";

        else if (firstOctet >= 128 && firstOctet <= 191) return "B";

        else if (firstOctet >= 192 && firstOctet <= 223) return "C";

        else return null;

    }

    static String decimalToBinary(String ip) {

        StringBuilder binary = new StringBuilder();

        for (String part : ip.split("\\.")) {

            int val = Integer.parseInt(part);

            binary.append(String.format("%8s", Integer.toBinaryString(val)).replace(' ', '0')).append(".");

        }

        return binary.substring(0, binary.length() - 1);

    }

    static String binaryToDecimal(String binaryIP) {

        StringBuilder decimal = new StringBuilder();

        for (String part : binaryIP.split("\\.")) {

            decimal.append(Integer.parseInt(part, 2)).append(".");

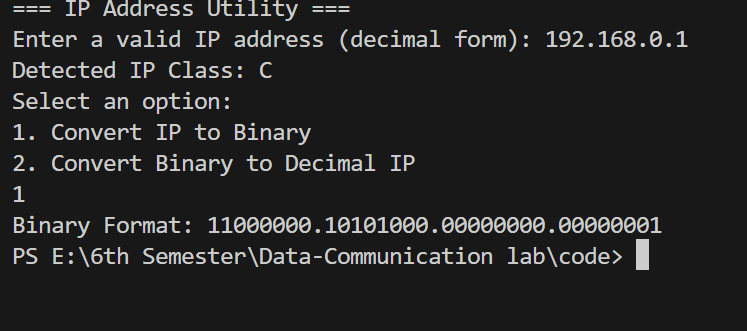
        }

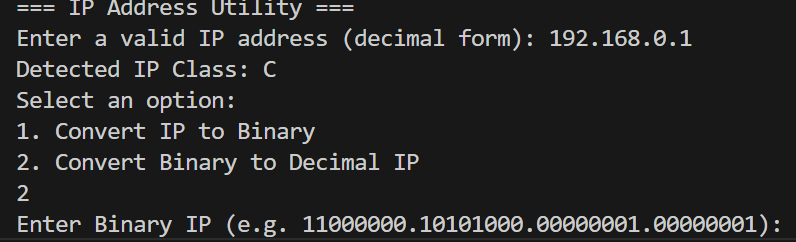
        return decimal.substring(0, decimal.length() - 1);

    }

}

**7.Output:**

****



### **8. Conclusion:**

This experiment provided hands-on experience with IP address classification and binary-decimal conversion logic. The program models how networks distinguish and represent IP addresses internally. Understanding such representation is critical in network addressing, routing, and subnetting.