

Code:-

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.Scanner;

public class IPAddress {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter The IP Address: ");
        String ipAddress = scanner.nextLine();

        try {
            InetAddress inetAddress = InetAddress.getByName(ipAddress);
            String[] ipParts = ipAddress.split("\\.");
            int firstOctet = Integer.parseInt(ipParts[0]);

            if (firstOctet == 10 || (firstOctet == 172 &&
Integer.parseInt(ipParts[1]) >= 16 && Integer.parseInt(ipParts[1]) <= 31)
|| (firstOctet == 192 && Integer.parseInt(ipParts[1]) ==
168)) {
                System.out.println("Given IP is Private");
            } else if (firstOctet == 127) {
                System.out.println("Given IP is Reserved for Loopback");
            } else if (firstOctet >= 1 && firstOctet <= 126) {
                System.out.println("Given IP is Public and belongs to Class
A");
            } else if (firstOctet >= 128 && firstOctet <= 191) {
                System.out.println("Given IP is Public and belongs to Class
B");
            } else if (firstOctet >= 192 && firstOctet <= 223) {
                System.out.println("Given IP is Public and belongs to Class
C");
            } else if (firstOctet >= 224 && firstOctet <= 239) {
                System.out.println("Given IP is Multicast");
            } else if (firstOctet >= 240 && firstOctet <= 255) {
                System.out.println("Given IP is Reserved for Future Use");
            } else {
                System.out.println("Given IP is Invalid");
            }

            char ipClass = getIPClass(firstOctet);
            System.out.println("This IP Address Belongs to Class: " +
ipClass);

            String defaultMask = getDefaultMask(ipClass);
            System.out.println("Default Mask: " + defaultMask);
        } catch (UnknownHostException e) {
            System.out.println("Invalid IP Address");
        }
    }

    private static char getIPClass(int firstOctet) {
        if (firstOctet <= 126) return 'A';
        if (firstOctet <= 191) return 'B';
        if (firstOctet <= 223) return 'C';
        if (firstOctet <= 239) return 'D';
        if (firstOctet <= 255) return 'E';
        return 'Invalid';
    }

    private static String getDefaultMask(char ipClass) {
        switch (ipClass) {
            case 'A': return "255.0.0.0";
            case 'B': return "255.255.0.0";
            case 'C': return "255.255.255.0";
            case 'D': return "240.0.0.0";
            case 'E': return "0.0.0.0";
            default: return "Invalid";
        }
    }
}
```

```

        String block = getBlock(ipAddress);
        System.out.println("Block: " + block);

        String firstIP = getFirstIP(inetAddress);
        System.out.println("First IP: " + firstIP);

        String lastIP = getLastIP(inetAddress);
        System.out.println("Last IP: " + lastIP);

        String networkAddress = getNetworkAddress(inetAddress);
        System.out.println("Network Address: " + networkAddress);

    } catch (NumberFormatException | UnknownHostException e) {
        System.out.println("Invalid IP address format");
    } finally {
        scanner.close();
    }
}

private static String getBlock(String ipAddress) {
    String[] parts = ipAddress.split("\\.");
    return parts[0];
}

private static String getFirstIP(InetAddress inetAddress) {
    byte[] ipBytes = inetAddress.getAddress();
    ipBytes[ipBytes.length - 1] = 1; // Set last octet to 1
    try {
        InetAddress firstIP = InetAddress.getByAddress(ipBytes);
        return firstIP.getHostAddress();
    } catch (UnknownHostException e) {
        e.printStackTrace();
        return "N/A";
    }
}

private static String getLastIP(InetAddress inetAddress) {
    byte[] ipBytes = inetAddress.getAddress();
    ipBytes[ipBytes.length - 1] = (byte) 0xFF; // Set last octet to 255
    try {
        InetAddress lastIP = InetAddress.getByAddress(ipBytes);
        return lastIP.getHostAddress();
    } catch (UnknownHostException e) {
        e.printStackTrace();
        return "N/A";
    }
}

```

```

private static String getNetworkAddress(InetAddress inetAddress) {
    byte[] ipBytes = inetAddress.getAddress();
    ipBytes[ipBytes.length - 1] = 0; // Set last octet to 0
    try {
        InetAddress networkAddress = InetAddress.getByAddress(ipBytes);
        return networkAddress.getHostAddress();
    } catch (UnknownHostException e) {
        e.printStackTrace();
        return "N/A";
    }
}

private static char getIPClass(int firstOctet) {
    if (firstOctet >= 1 && firstOctet <= 126) {
        return 'A';
    } else if (firstOctet >= 128 && firstOctet <= 191) {
        return 'B';
    } else if (firstOctet >= 192 && firstOctet <= 223) {
        return 'C';
    } else if (firstOctet >= 224 && firstOctet <= 239) {
        return 'D';
    } else {
        return 'E';
    }
}

private static String getDefaultMask(char ipClass) {
    switch (ipClass) {
        case 'A':
            return "255.0.0.0";
        case 'B':
            return "255.255.0.0";
        case 'C':
            return "255.255.255.0";
        default:
            return "N/A";
    }
}
}

```

Output:

```
● PS C:\Users\91842\OneDrive\Desktop\codes> cd "c:\Users\91842\OneDrive\Desktop\codes"
a IPAddress }
Enter The IP Address: 252.78.98.90
Given IP is Reserved for Future Use
This IP Address Belongs to Class: E
Default Mask: N/A
Block: 252
First IP: 252.78.98.1
Last IP: 252.78.98.255
Network Address: 252.78.98.0
```