



## Experiment No: 07

**Name:** Shashwat Shah

**SAP ID:** 60004220126

**Aim:** Write a program to identify the class and subnet address of the given IP address.

### Theory:

Internet addresses are allocated by the InterNIC, the organization that administers the Internet. These IP addresses are divided into classes. The most common of them are classes A, B, and C. Classes D and E exist, but aren't used by end users. Each of the address classes has a different default subnet mask. You can identify the class of an IP address by looking at its first octet. Following are the ranges of Class A, B, and C Internet addresses, each with an example address:

- Class A networks use a default subnet mask of 255.0.0.0 and have 0-127 as their first octet. The address 10.52.36.11 is a class A address. Its first octet is 10, which is between 1 and 126, inclusive.
- Class B networks use a default subnet mask of 255.255.0.0 and have 128-191 as their first octet. The address 172.16.52.63 is a class B address. Its first octet is 172, which is between 128 and 191, inclusive.
- Class C networks use a default subnet mask of 255.255.255.0 and have 192-223 as their first octet. The address 192.168.123.132 is a class C address. Its first octet is 192, which is between 192 and 223, inclusive.

In some scenarios, the default subnet mask values don't fit the organization needs for one of the following reasons:

- The physical topology of the network
- The numbers of networks (or hosts) don't fit within the default subnet mask restrictions

### Subnetting:

- A Class A, B, or C TCP/IP network can be further divided, or sub netted, by a system administrator. It becomes necessary as you reconcile the logical address scheme of the Internet (the abstract world of IP addresses and subnets) with the physical networks in use by the real world.
- A system administrator who is allocated a block of IP addresses may be administering networks that aren't organized in a way that easily fits these addresses. For example, you have a wide area network with 150 hosts on three networks (in different cities) that are connected by a TCP/IP router. Each of these three networks has 50 hosts. You are allocated the class C network 192.168.123.0. (For illustration, this address is actually from a range that isn't allocated on the Internet.) It means that you can use the addresses 192.168.123.1 to 192.168.123.254 for your 150 hosts.
- Two addresses that can't be used in your example are 192.168.123.0 and 192.168.123.255 because binary addresses with a host portion of all ones and all zeros are invalid. The zero address is invalid because it's used to specify a network without specifying a host.



The 255 addresses (in binary notation, a host address of all ones) is used to broadcast a message to every host on a network. Just remember that the first and last address in any network or subnet can't be assigned to any individual host.

- You should now be able to give IP addresses to 254 hosts. It works fine if all 150 computers are on a single network. However, your 150 computers are on three separate physical networks. Instead of requesting more address blocks for each network, you divide your network into subnets that enable you to use one block of addresses on multiple physical networks.
- In this case, you divide your network into four subnets by using a subnet mask that makes the network address larger and the possible range of host addresses smaller. In other words, you are 'borrowing' some of the bits used for the host address, and using them for the network portion of the address. The subnet mask 255.255.255.192 gives you four networks of 62 hosts each. It works because in binary notation, 255.255.255.192 is the same as 11111111.11111111.11111111.11000000. The first two digits of the last octet become network addresses, so you get the additional networks 00000000 (0), 01000000 (64), 10000000 (128) and 11000000 (192). (Some administrators will only use two of the subnetworks using 255.255.255.192 as a subnet mask. For more information on this topic, see RFC 1878.) In these four networks, the last six binary digits can be used for host addresses.
- Using a subnet mask of 255.255.255.192, your 192.168.123.0 network then becomes the four networks 192.168.123.0, 192.168.123.64, 192.168.123.128 and 192.168.123.192. These four networks would have as valid host addresses:
- 192.168.123.1-62 192.168.123.65-126 192.168.123.129-190 192.168.123.193-254
- Remember, again, that binary host addresses with all ones or all zeros are invalid, so you can't use addresses with the last octet of 0, 63, 64, 127, 128, 191, 192, or 255.
- You can see how it works by looking at two host addresses, 192.168.123.71 and 192.168.123.133. If you used the default Class C subnet mask of 255.255.255.0, both addresses are on the 192.168.123.0 network. However, if you use the subnet mask of 255.255.255.192, they are on different networks; 192.168.123.71 is on the 192.168.123.64 network, 192.168.123.133 is on the 192.168.123.128 network.

#### Code:

```
import java.io.*;
```

```
// import java.net.InetAddress;
```

```
public class subnet{
```

```
    public static void main(String[] args) throws IOException{
```

```
        System.out.println("Enter IP Address: ");
```

```
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

```
        String ip = br.readLine();
```

```
        String checkclass = ip.substring(0,3);
```

```
        int cc = Integer.parseInt(checkclass);
```

```
        String mask = null;
```

```
        if(cc>0 && cc<224){
```

```
            if(cc<=127){
```



```
System.out.println("IP Address is of Class A");
mask="255.0.0.0";
}
if(cc>=128 && cc<=191){
    System.out.println("IP Address is of Class B");
    mask="255.255.0.0";
}
if(cc>=192 && cc<=223){
    System.out.println("IP Address is of Class C");
    mask="255.255.255.0";
}
if(cc>=224){
    System.out.println("IP Address is used for Multicasting or reserved");
}
}
System.out.println("Subnet Mask:\n"+mask);
String networkAddr="";

String[] ipAddrParts = ip.split("\\.");
String[] maskParts = mask.split("\\.");

for(int i=0;i<=3;i++){
    int x = Integer.parseInt(ipAddrParts[i]);
    int y = Integer.parseInt(maskParts[i]);

    int z = x & y;

    networkAddr += z+".";
}
System.out.println("SUBNET Address:\n"+networkAddr);
}
}
```

### Output:

```
Enter IP Address:
187.32.13.12
IP Address is of Class B
Subnet Mask:
255.255.0.0
SUBNET Address:
187.32.0.0.
```

### Conclusion:

The program to identify the class and subnet address of the given IP address is implemented.