Title: Write a program to find class and the type of the given IP address

1.InetAddress Classes:

- The `InetAddress` class is part of the `java.net` package.
- It provides methods for creating instances of InetAddress and obtaining information about hosts identified by IP addresses or domain names.
- InetAddress instances can represent both IPv4 and IPv6 addresses.

2. IP Address Classes:

- IP addresses are categorized into different classes based on the range of the first octet. There are five classes: A, B, C, D, and E.
- **Class A:** The first octet is the network portion, and the remaining three octets are for hosts. Range: 1.0.0.0 to 126.255.255.255.
- **Class B:** The first two octets are the network portion, and the remaining two octets are for hosts. Range: 128.0.0.0 to 191.255.255.255.
- **Class C:** The first three octets are the network portion, and the last octet is for hosts. Range: 192.0.0.0 to 223.255.255.255.
- **Class D:** Reserved for multicast groups. Range: 224.0.0.0 to 239.255.255.255.
- **Class E:** Reserved for experimental purposes. Range: 240.0.0.0 to 255.255.255.255.

3. IP Address Types:

- InetAddress also provides information about the type or scope of an IP address. Some common types include:
  - **Loopback Address:** Refers to the local machine and is typically represented as `127.0.0.1` in IPv4.
  - **Link Local Address:** Used for communication within the local network segment.
  - **Site Local Address:** Used for communication within a specific site or organization.
  - **Multicast Address:** Identifies a group of hosts that can be reached simultaneously.
  - **Global Address:** Typically used for addressing on the global Internet.
  - **Node Local, Organization Local, and Site Local Multicast:** Specific multicast scopes.

4. IPv6 Consideration:

- While the original concepts of IP address classes are based on IPv4, with the introduction of IPv6, the addressing scheme becomes more flexible and is not strictly tied to classes.

Code:

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

public class IPAddressInfo {
public static void main(String[] args) {
String ipAddress = "192.168.1.1";

try {
InetAddress inetAddress = InetAddress.getByName(ipAddress);

char ipClass = getIPClass(inetAddress);

String ipType = getIPType(inetAddress);

System.out.println("IP Address: " + inetAddress.getHostAddress());
System.out.println("IP Class: " + ipClass);
System.out.println("IP Type: " + ipType);

} catch (UnknownHostException e) {
System.out.println("Invalid IP Address: " + ipAddress);
}
}

private static char getIPClass(InetAddress inetAddress) {
byte[] ipAddressBytes = inetAddress.getAddress();
int firstOctet = Byte.toUnsignedInt(ipAddressBytes[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 if (firstOctet >= 224 && firstOctet <= 239) {
return 'D';
```

```
} else if (firstOctet >= 240 && firstOctet <= 255) {
return 'E';
} else {
return ' ';
}
}

private static String getIPType(InetAddress inetAddress) {
if (inetAddress.isAnyLocalAddress()) {
return "Any Local Address";
} else if (inetAddress.isLinkLocalAddress()) {
return "Link Local Address";
} else if (inetAddress.isLoopbackAddress()) {
return "Loopback Address";
} else if (inetAddress.isMulticastAddress()) {
return "Multicast Address";
} else if (inetAddress.isSiteLocalAddress()) {
return "Site Local Address";
} else if (inetAddress.isMCGlobal()) {
return "Multicast Global";
} else if (inetAddress.isMCLinkLocal()) {
return "Multicast Link Local";
} else if (inetAddress.isMCNodeLocal()) {
return "Multicast Node Local";
} else if (inetAddress.isMCOrgLocal()) {
return "Multicast Organization Local";
} else if (inetAddress.isMCSiteLocal()) {
return "Multicast Site Local";
} else {
return "Unknown Type";
}
}
}
```

Output: