1. Write a JAVA program to illustrate different types of address.

```
import java.net.*;
public class DiffAddress {
public static void main(String[] args) {
try {
//Localhost Address (IP of current machine)
InetAddress localHost = InetAddress.getLocalHost();
System.out.println("Local Host Address: " + localHost);
//Loopback Address (Refers to machine itself)
InetAddress loopback = InetAddress.getLoopbackAddress();
System.out.println("Loopback Address: " + loopback);
//IP Address of a Domain (e.g., Google)
InetAddress google = InetAddress.getByName("www.google.com");
System.out.println("Google's IP Address: " + google.getHostAddress());
//All IP Addresses of a Domain (Microsoft)
InetAddress[] addresses = InetAddress.getAllByName("www.microsoft.com");
System.out.println("Microsoft IP Addresses:");
for (InetAddress address : addresses) {
System.out.println(" - " + address.getHostAddress());
}
} catch (UnknownHostException e) {
System.out.println("Error: " + e.getMessage());
}
}
}
```

2. In what ways getHostName() differs from getCannonicalHostName()?

getHostName() :The getHostName() method returns the hostname associated with an IP address. If the hostname is not available it returns the IP address as a string. This method often uses cached information or immediate system resources, meaning it may not perform a DNS lookup. Because of this, it’s faster but less reliable for getting the full or official hostname.

getCanonicalHostName() :The getCanonicalHostName() method tries to return the fully qualified domain name (FQDN) for the IP address.This method performs a reverse DNS lookup, making it more accurate but it is slower. If successful, it returns the official domain name associated with the IP.

Write a JAVA program that, Displays both hostname and the cannonical hostname of a domain.

```
import java.net.*;
public class HostnameChecker {
 public static void main(String[] args) {
 try {
String domain ="www.google.com";
InetAddress address = InetAddress.getByName(domain);
// Display hostname and canonical hostname
System.out.println("Domain Name: " + domain);//OP-Domain Name: www.google.com
System.out.println("Host Name: " + address.getHostName());//OP-Host Name:www.google.com
System.out.println("Canonical Host Name: " + address.getCanonicalHostName());
//OP-Canonical Host Name: del03s29-in-f4.1e100.net
} catch (UnknownHostException e) {
System.out.println("Unable to resolve the domain. Error: " + e.getMessage());
}
}
}
```

Write a JAVA program that,Utilizes getAddress() to determine if the given address is IPv4 or IPv6.

```
import java.net.*;
import java.util.Scanner;
public class IPVersionChecker {
public static void main(String[] args) {
 try (Scanner scanner= new Scanner(System.in)) {
System.out.print("Enter a domain or IP address: ");
String input= scanner.nextLine();
```

```
// Get InetAddress object
InetAddress address = InetAddress.getByName(input);
// raw IP address in byte array form
byte[] ip = address.getAddress();
// Check IP version based on byte array length
if (ip.length == 4) {
System.out.println("The address is IPv4.");
} else if (ip.length == 16) {
System.out.println("The address is IPv6.");
} else {
System.out.println("Unknown IP version.");
}
// print resolved IP address
System.out.println("Resolved IP: " + address.getHostAddress());
} catch (UnknownHostException e) {
System.out.println("Invalid address. Error: " + e.getMessage());
}
}
}
```