# Template Week 6 – Networking

Student number:568524

**Assignment 6.1: Working from home**

Screenshot installation openssh-server:

Screenshot successful SSH command execution:

Screenshot successful execution SCP command:

Screenshot remmina:

**Assignment 6.2: IP addresses websites**

Relevant screenshots nslookup command:

Screenshot website visit via IP address:

**Assignment 6.3: subnetting**

How many IP addresses are in this network configuration 192.168.110.128/25?

What is the usable IP range to hand out to the connected computers?

Check your two previous answers with this calculator:
https://www.calculator.net/ip-subnet-calculator.html

Explain the above calculation in your own words.

**Assignment 6.4: HTML**

Screenshot IP address Ubuntu VM:

Screenshot of Site directory contents:

Screenshot python3 webserver command:

Screenshot web browser visits your site

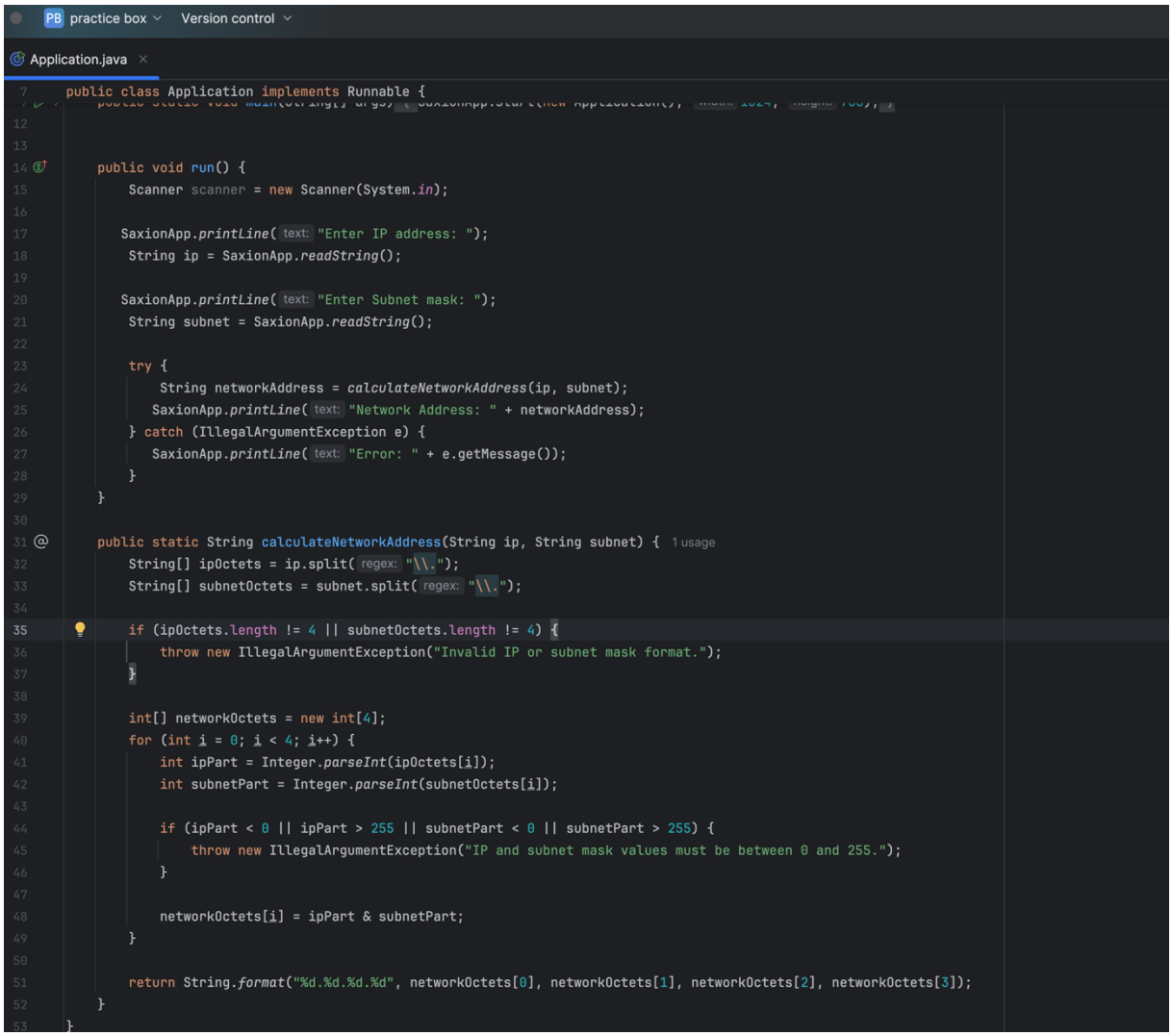
**Bonus point assignment – week 6**

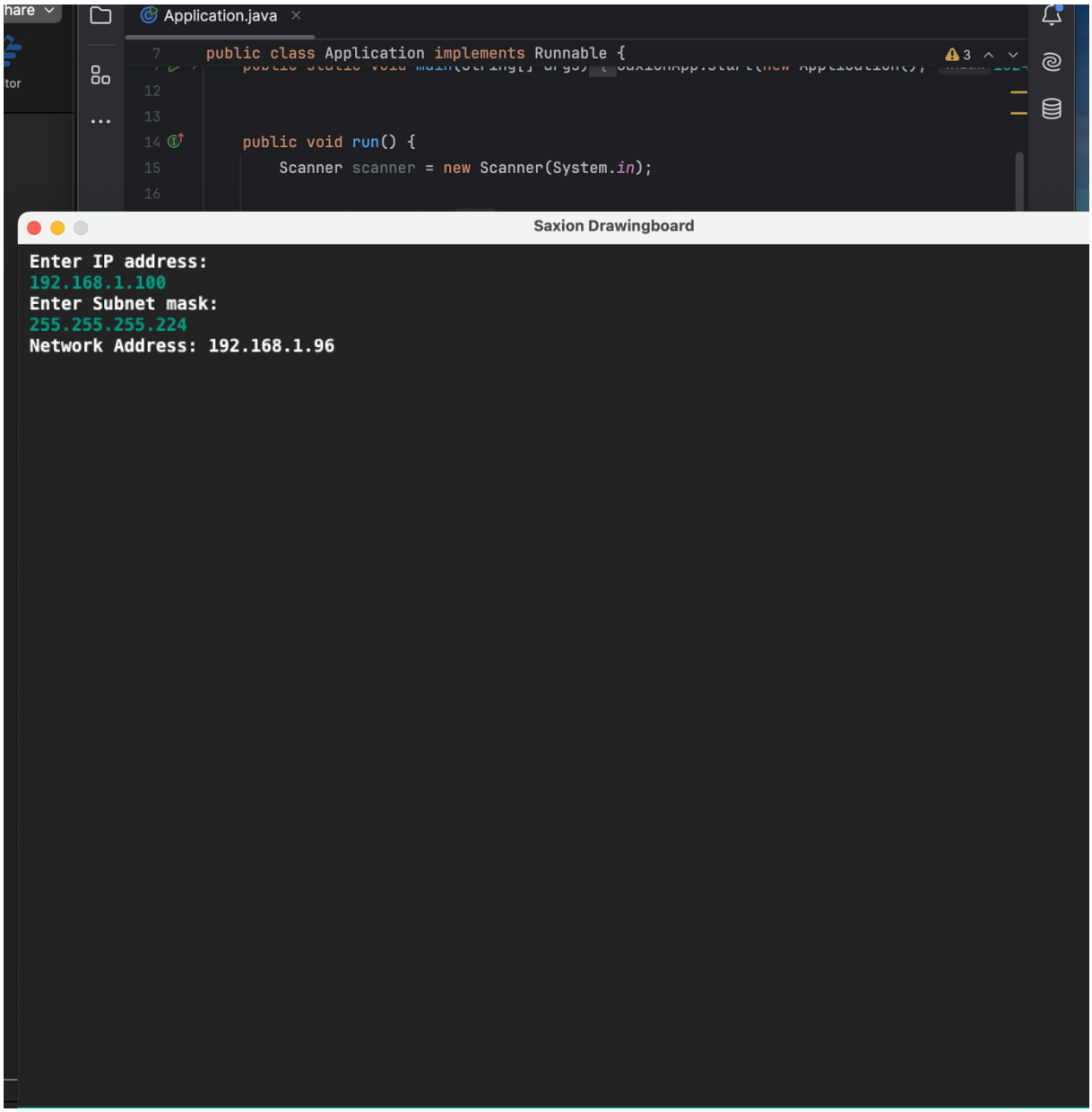
Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

```
Example: 192.168.1.100/27
Calculate the network segment
IP Address:   11000000.10101000.00000001.01100100
Subnet Mask:  11111111.11111111.11111111.11100000
-----------------------------------------------
Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.
For a /27 subnet, each segment (or subnet) has 32 IP addresses (2^5).
The range of this network segment is from 192.168.1.96 to 192.168.1.127.
```


practice box
Version control
Application.java
public class Application implements Runnable {
public void run() {
Scanner scanner = new Scanner(System.in);
SaxionApp.printLine( text: "Enter IP address: ");
String ip = SaxionApp.readString();
SaxionApp.printLine( text: "Enter Subnet mask: ");
String subnet = SaxionApp.readString();
try {
String networkAddress = calculateNetworkAddress(ip, subnet);
SaxionApp.printLine( text: "Network Address: " + networkAddress);
} catch (IllegalArgumentException e) {
SaxionApp.printLine( text: "Error: " + e.getMessage());
}
}
public static String calculateNetworkAddress(String ip, String subnet) { 1 usage
String[] ipOctets = ip.split( regex: "\\.");
String[] subnetOctets = subnet.split( regex: "\\.");
if (ipOctets.length != 4 || subnetOctets.length != 4) {
throw new IllegalArgumentException("Invalid IP or subnet mask format.");
}
int[] networkOctets = new int[4];
for (int i = 0; i < 4; i++) {
int ipPart = Integer.parseInt(ipOctets[i]);
int subnetPart = Integer.parseInt(subnetOctets[i]);
if (ipPart < 0 || ipPart > 255 || subnetPart < 0 || subnetPart > 255) {
throw new IllegalArgumentException("IP and subnet mask values must be between 0 and 255.");
}
networkOctets[i] = ipPart & subnetPart;
}
return String.format("%d.%d.%d.%d", networkOctets[0], networkOctets[1], networkOctets[2], networkOctets[3]);
}
}




Paste source code here, with a screenshot of a working application.

Ready? Save this file and export it as a pdf file with the name: **week6.pdf**