Template Week 6 – Networking

Student number:
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:
Screenshot Website visit via ir 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.

IT FUNDAMENTALS 1

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

Example: 192.168.1.100/27

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

```
Calculate the network segment
IP Address: 11000000.10101000.00000001.01100100
Subnet Mask: 1111111.1111111.11111111.1111111.111100000
-------
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<sup>5</sup>).
The range of this network segment is from 192.168.1.96 to 192.168.1.127.

Paste source code here, with a screenshot of a working application.
import java.util.Scanner;
```

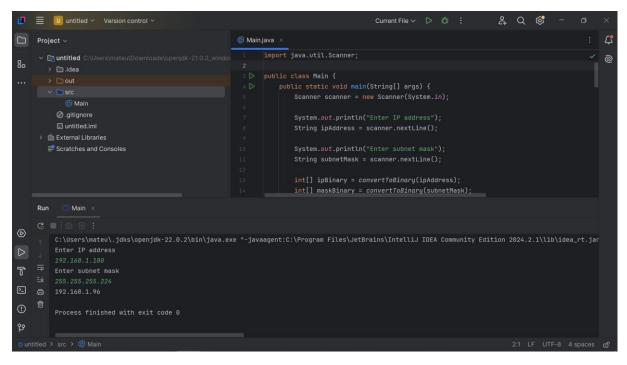
IT FUNDAMENTALS

2

```
System.out.println("Enter IP address");
  String ipAddress = scanner.nextLine();
  System.out.println("Enter subnet mask");
  String subnetMask = scanner.nextLine();
  int[] ipBinary = convertToBinary(ipAddress);
  int[] maskBinary = convertToBinary(subnetMask);
  if (ipBinary == null | | maskBinary == null) {
    System.out.println("Something went wrong. Please try again.");
    return;
  }
  int[] networkAddress = calculateNetworkAddress(ipBinary, maskBinary);
  String networkAddressDecimal = convertToDecimal(networkAddress);
  System.out.println(networkAddressDecimal);
}
public static int[] convertToBinary(String address) {
  String[] parts = address.split("\\.");
  if (parts.length != 4) {
    return null;
  int[] binary = new int[32];
  try {
    for (int i = 0; i < 4; i++) {
      int octet = Integer.parseInt(parts[i]);
      String binaryString = String.format("%8s", Integer.toBinaryString(octet)).replace('', '0');
       for (int j = 0; j < 8; j++) {
         binary[i * 8 + j] = binaryString.charAt(j) - '0';
      }
  } catch (NumberFormatException e) {
    return null;
  }
  return binary;
}
public static int[] calculateNetworkAddress(int[] ip, int[] mask) {
  int[] network = new int[32];
  for (int i = 0; i < 32; i++) {
    network[i] = ip[i] & mask[i];
  }
  return network;
}
```

IT FUNDAMENTALS 3

```
public static String convertToDecimal(int[] binary) {
    StringBuilder decimalAddress = new StringBuilder();
    for (int i = 0; i < 4; i++) {
        int octet = 0;
        for (int j = 0; j < 8; j++) {
            octet = (octet << 1) | binary[i * 8 + j];
        }
        decimalAddress.append(octet);
        if (i < 3) {
            decimalAddress.append(".");
        }
    }
    return decimalAddress.toString();
}</pre>
```



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

IT FUNDAMENTALS 4