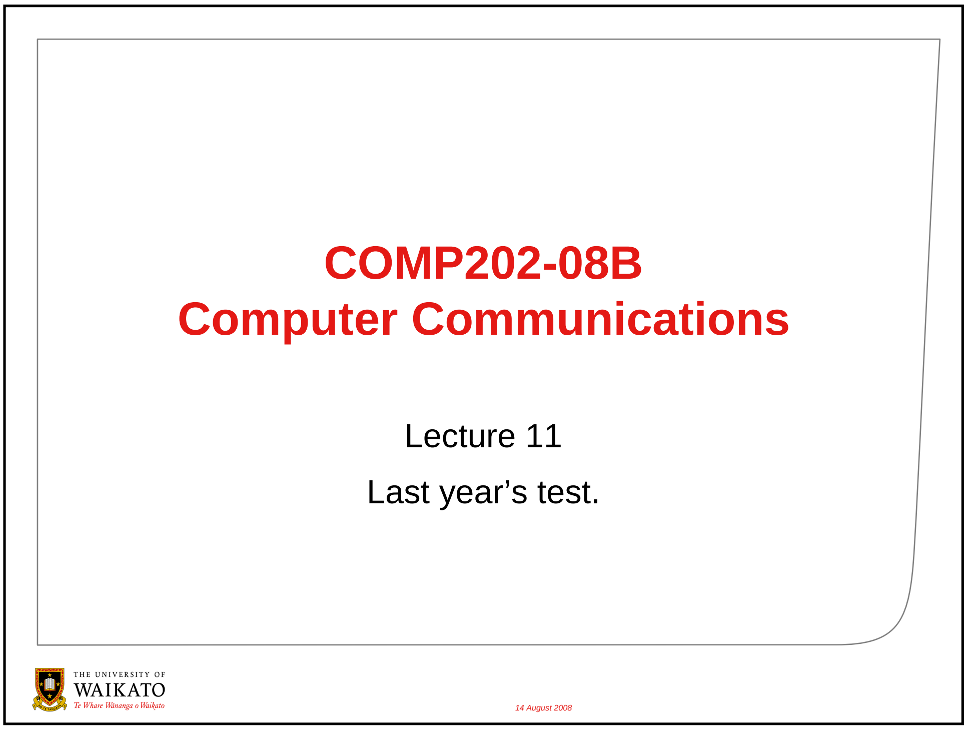
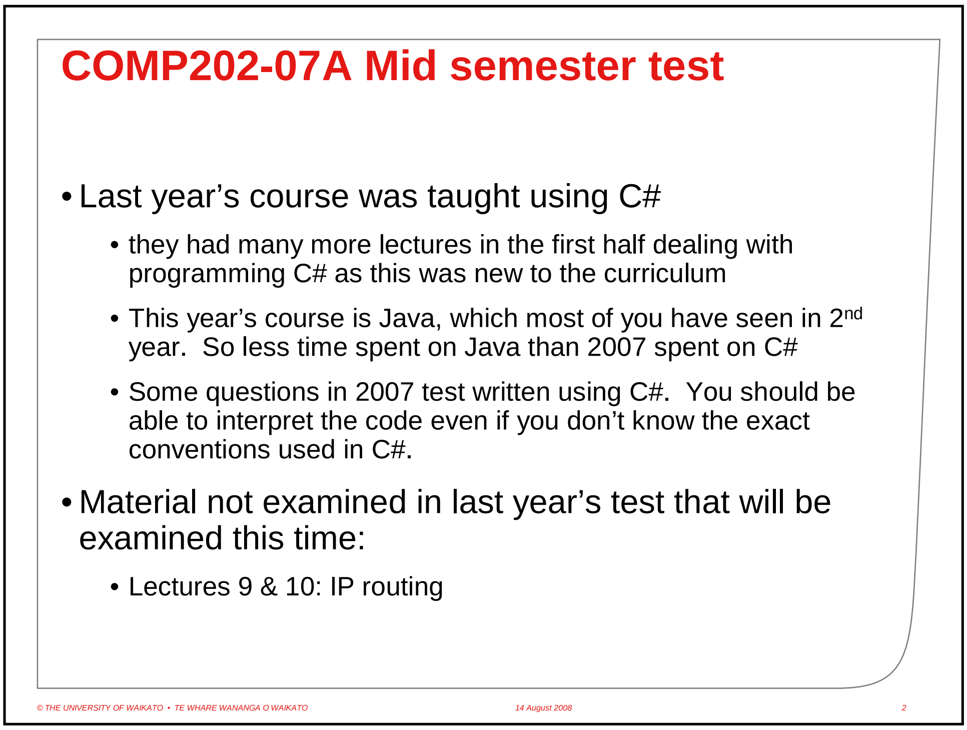


COMP202-08B
Computer Communications

Lecture 11

Last year's test.

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato



COMP202-07A Mid semester test

- Last year's course was taught using C#
  - they had many more lectures in the first half dealing with programming C# as this was new to the curriculum
  - This year's course is Java, which most of you have seen in 2nd year. So less time spent on Java than 2007 spent on C#
  - Some questions in 2007 test written using C#. You should be able to interpret the code even if you don't know the exact conventions used in C#.
- Material not examined in last year's test that will be examined this time:
  - Lectures 9 & 10: IP routing

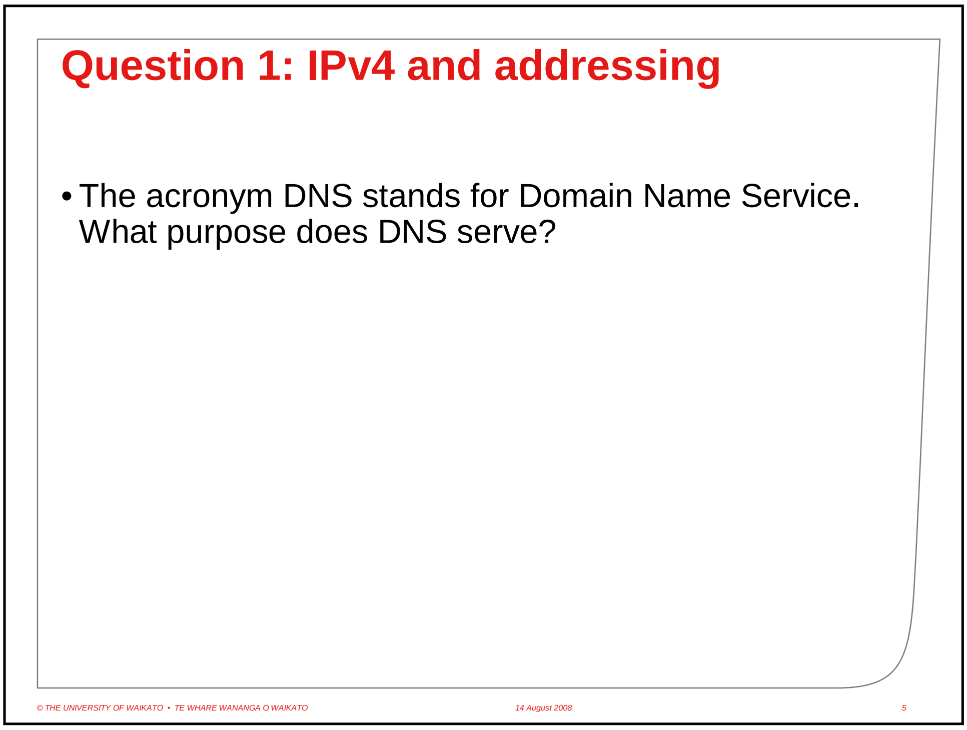
Question 1: IPv4 and addressing

- An ‘End point’ is made up of an ‘IP address’ and a ‘Port number’. What do these terms mean and what is the purpose of each?
  - End point:
  - IP address:
  - Port number:

Question 1: IPv4 and addressing

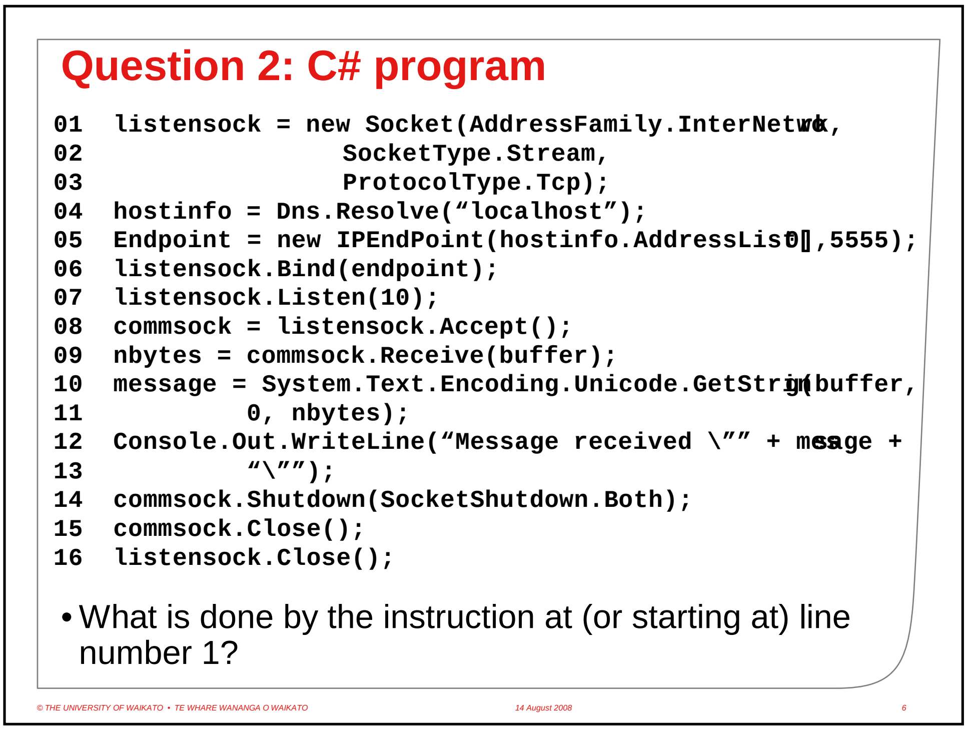
- A machine usually has a ‘Domain’ name as well as an IP address. To what does the domain name ‘localhost’ refer and what IP address is (usually) used for ‘localhost’?

- Note: we have not explicitly covered localhost in 2008. Try and answer the question anyway.



Question 1: IPv4 and addressing

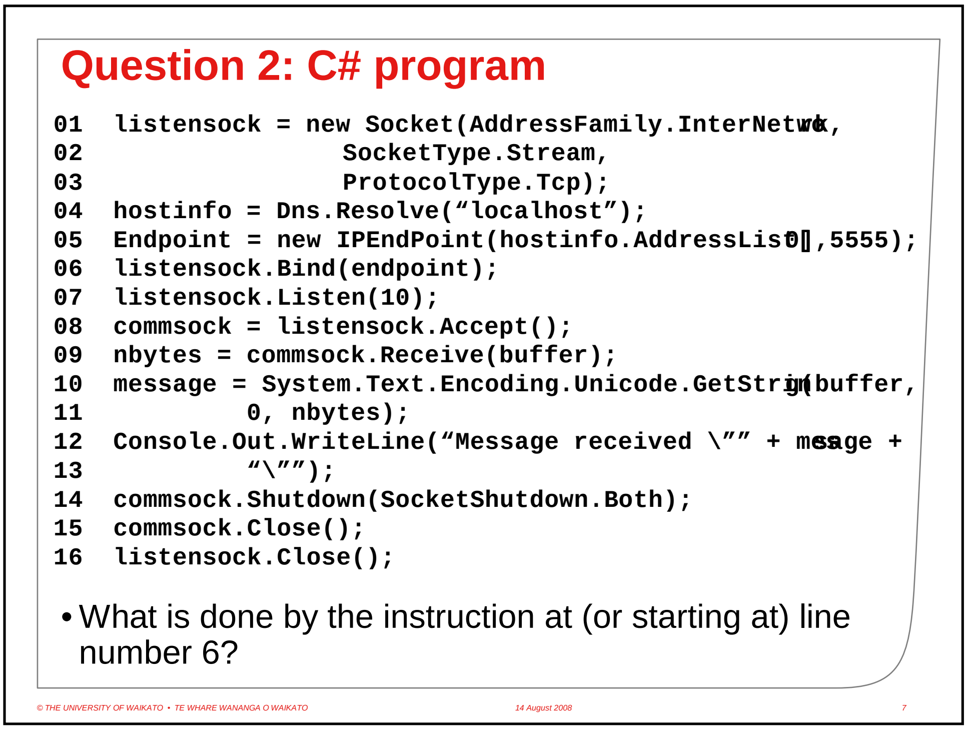
- The acronym DNS stands for Domain Name Service. What purpose does DNS serve?



Question 2: C# program

```
01  listensock = new Socket(AddressFamily.InterNetwork,
02                 SocketType.Stream,
03                 ProtocolType.Tcp);
04  hostinfo = Dns.Resolve(“localhost”);
05  Endpoint = new IPEndPoint(hostinfo.AddressList[0],5555);
06  listensock.Bind(endpoint);
07  listensock.Listen(10);
08  commsock = listensock.Accept();
09  nbytes = commsock.Receive(buffer);
10  message = System.Text.Encoding.Unicode.GetString(buffer,
11           0, nbytes);
12  Console.Out.WriteLine(“Message received \”” + message +
13           “\””);
14  commsock.Shutdown(SocketShutdown.Both);
15  commsock.Close();
16  listensock.Close();
```

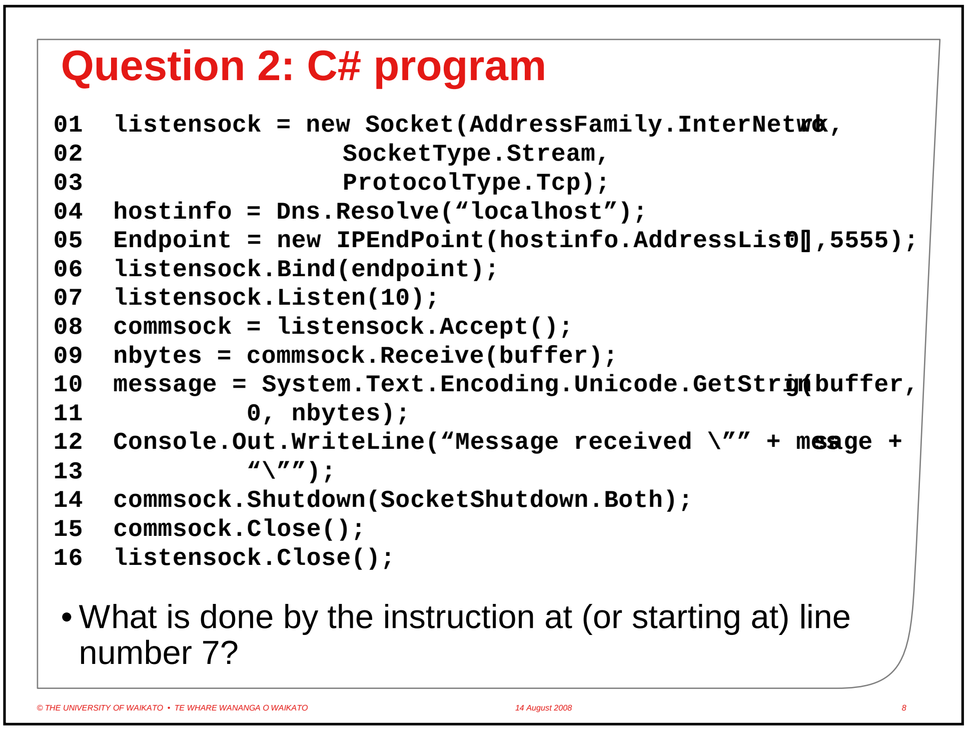
- What is done by the instruction at (or starting at) line number 1?



Question 2: C# program

```
01  listensock = new Socket(AddressFamily.InterNetwork,
02                     SocketType.Stream,
03                     ProtocolType.Tcp);
04  hostinfo = Dns.Resolve("localhost");
05  Endpoint = new IPEndPoint(hostinfo.AddressList[0],5555);
06  listensock.Bind(endpoint);
07  listensock.Listen(10);
08  commsock = listensock.Accept();
09  nbytes = commsock.Receive(buffer);
10  message = System.Text.Encoding.Unicode.GetString(buffer,
11            0, nbytes);
12  Console.Out.WriteLine("Message received \"" + message +
13            "\"");
14  commsock.Shutdown(SocketShutdown.Both);
15  commsock.Close();
16  listensock.Close();
```

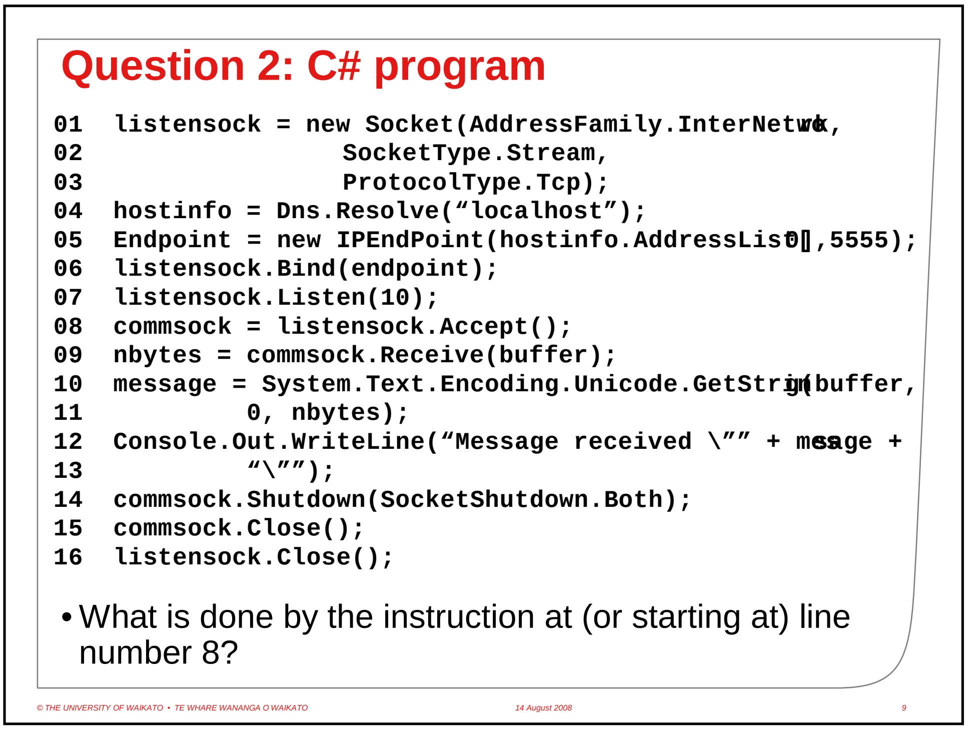
- What is done by the instruction at (or starting at) line number 6?



Question 2: C# program

```
01  listensock = new Socket(AddressFamily.InterNetwork,
02                     SocketType.Stream,
03                     ProtocolType.Tcp);
04  hostinfo = Dns.Resolve("localhost");
05  Endpoint = new IPEndPoint(hostinfo.AddressList[0],5555);
06  listensock.Bind(endpoint);
07  listensock.Listen(10);
08  commsock = listensock.Accept();
09  nbytes = commsock.Receive(buffer);
10  message = System.Text.Encoding.Unicode.GetString(buffer,
11            0, nbytes);
12  Console.Out.WriteLine("Message received \"" + message +
13            "\"");
14  commsock.Shutdown(SocketShutdown.Both);
15  commsock.Close();
16  listensock.Close();
```

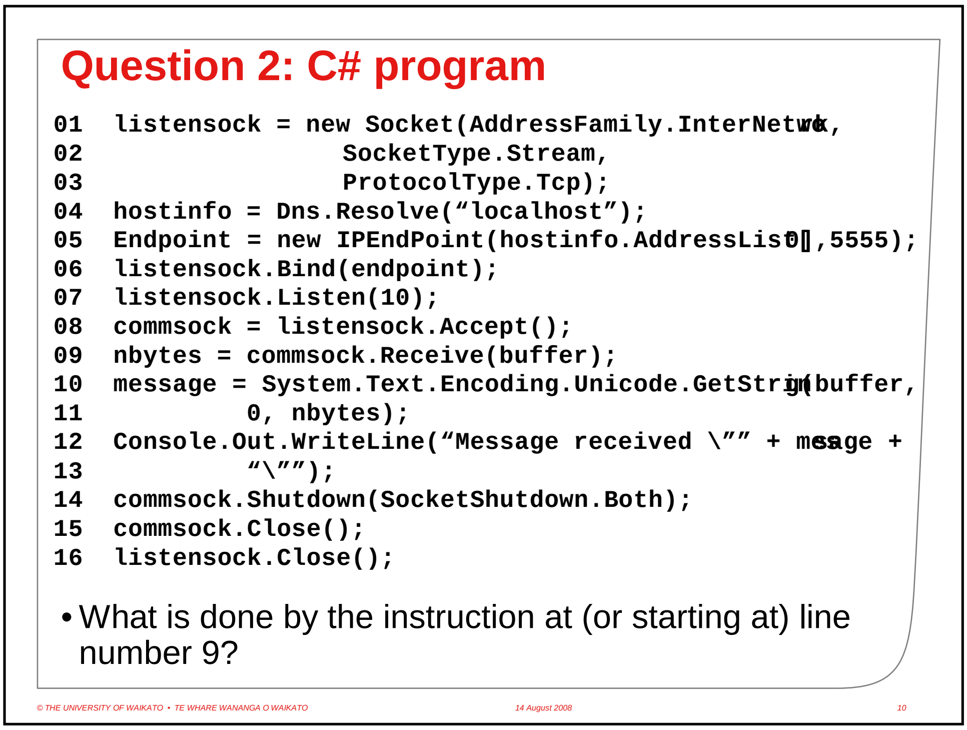
- What is done by the instruction at (or starting at) line number 7?



Question 2: C# program

```
01  listensock = new Socket(AddressFamily.InterNetwork,
02                 SocketType.Stream,
03                 ProtocolType.Tcp);
04  hostinfo = Dns.Resolve(“localhost”);
05  Endpoint = new IPEndPoint(hostinfo.AddressList[0],5555);
06  listensock.Bind(endpoint);
07  listensock.Listen(10);
08  commsock = listensock.Accept();
09  nbytes = commsock.Receive(buffer);
10  message = System.Text.Encoding.Unicode.GetString(buffer,
11           0, nbytes);
12  Console.Out.WriteLine(“Message received \”” + message +
13           “\””);
14  commsock.Shutdown(SocketShutdown.Both);
15  commsock.Close();
16  listensock.Close();
```

- What is done by the instruction at (or starting at) line number 8?



Question 2: C# program

```
01  listensock = new Socket(AddressFamily.InterNetwork,
02                 SocketType.Stream,
03                 ProtocolType.Tcp);
04  hostinfo = Dns.Resolve(“localhost”);
05  Endpoint = new IPEndPoint(hostinfo.AddressList[0],5555);
06  listensock.Bind(endpoint);
07  listensock.Listen(10);
08  commsock = listensock.Accept();
09  nbytes = commsock.Receive(buffer);
10  message = System.Text.Encoding.Unicode.GetString(buffer,
11           0, nbytes);
12  Console.Out.WriteLine(“Message received \”” + message +
13           “\””);
14  commsock.Shutdown(SocketShutdown.Both);
15  commsock.Close();
16  listensock.Close();
```

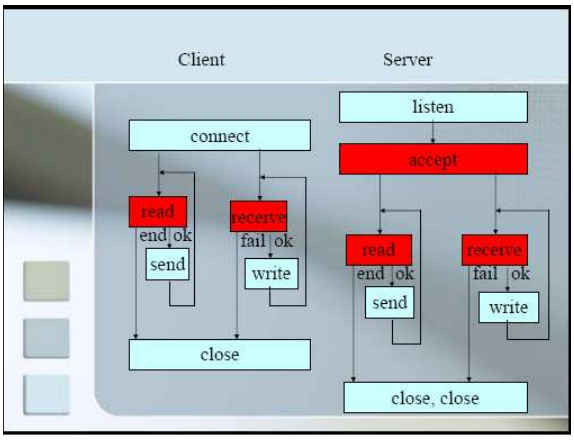
- What is done by the instruction at (or starting at) line number 9?

Question 3: Threads

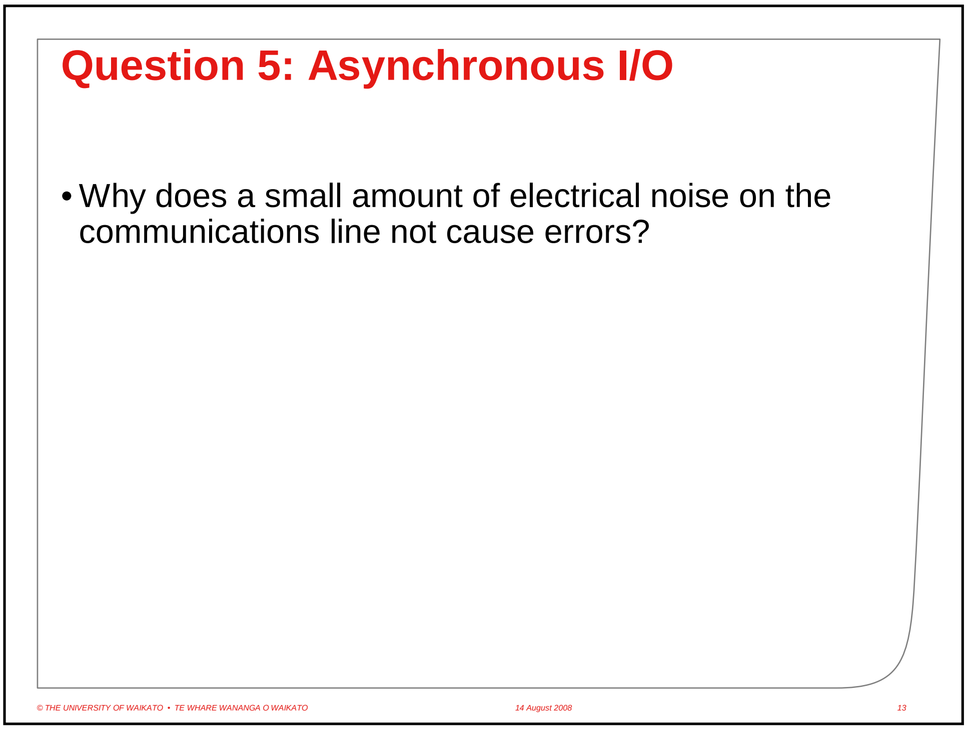
```
static void Sleepy() {
  Thread.Sleep(2000);
  Console.Out.WriteLine(“Sleepy A”);
  Thread.Sleep(2000);
  Console.Out.WriteLine(“Sleepy B”);
}
static void Snoozy() {
  Thread.Sleep(1500);
  Console.Out.WriteLine(“Snoozy X”);
  Thread.Sleep(3000);
  Console.Out.WriteLine(“Snoozy Y”);
}
static void Main() {
  Thread one = new Thread(new ThreadStart(Sleepy));
  one.start();
  Thread two = new Thread(new ThreadStart(Snoozy));
  two.start();
  Console.Out.WriteLine(“Main”);
}
```

What output would you expect to have displayed when the program is run (starting at Main). You are required to show the lines of output in the correct order.

Question 4: Client and Server programs

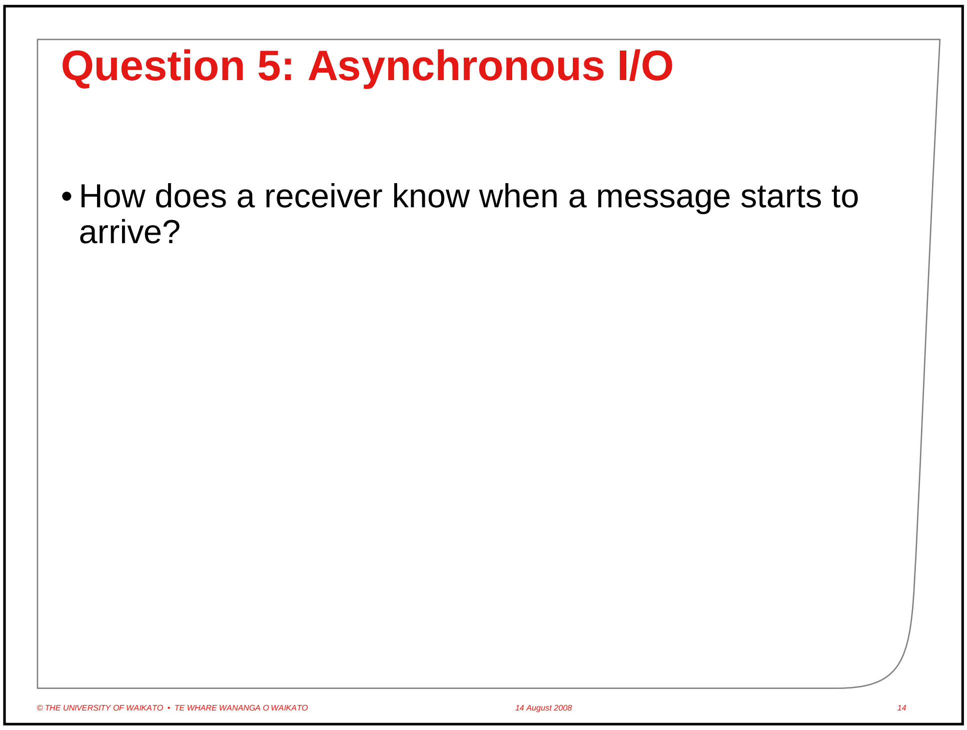


- Explain why and for what purpose we use threads in these programs. In particular you should explain what program behaviour we can get using threads that would otherwise be difficult to achieve.



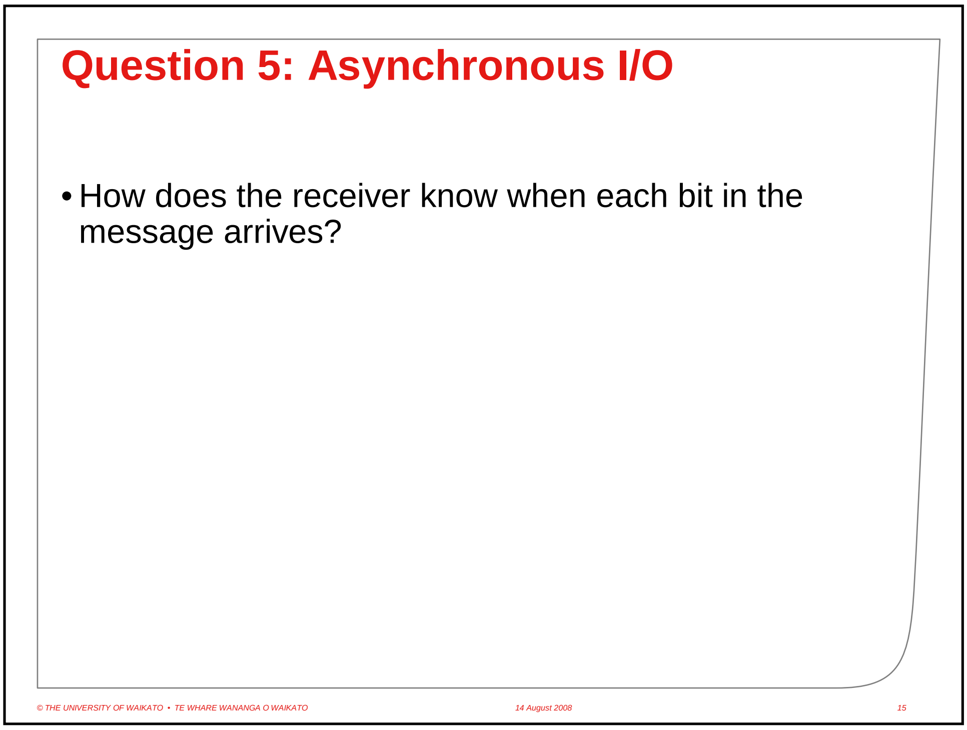
Question 5: Asynchronous I/O

- Why does a small amount of electrical noise on the communications line not cause errors?



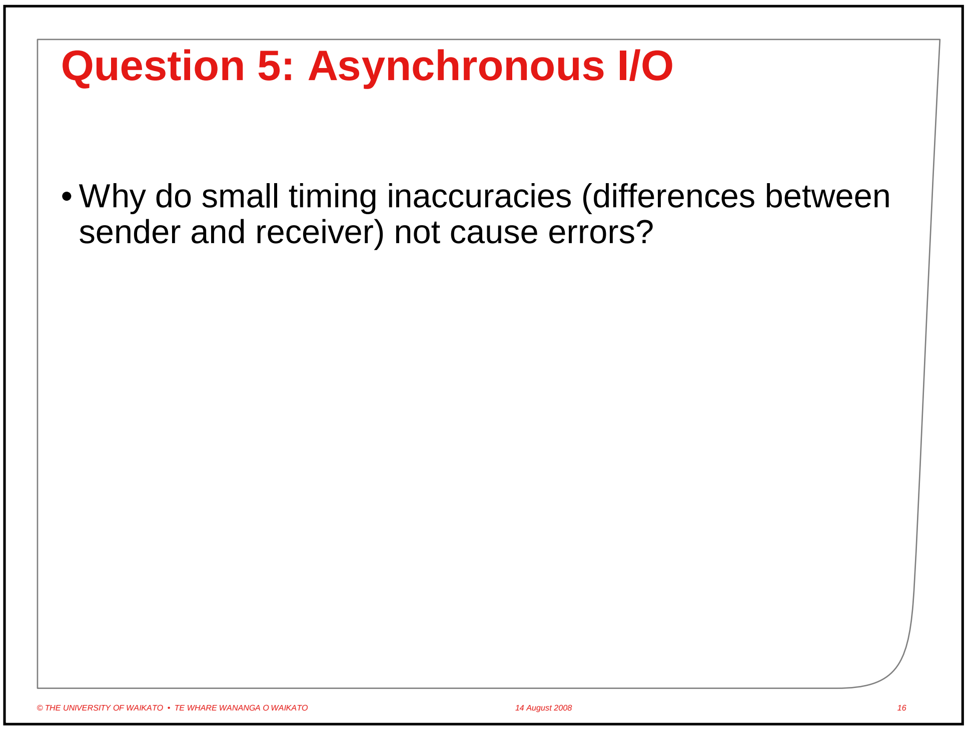
Question 5: Asynchronous I/O

- How does a receiver know when a message starts to arrive?



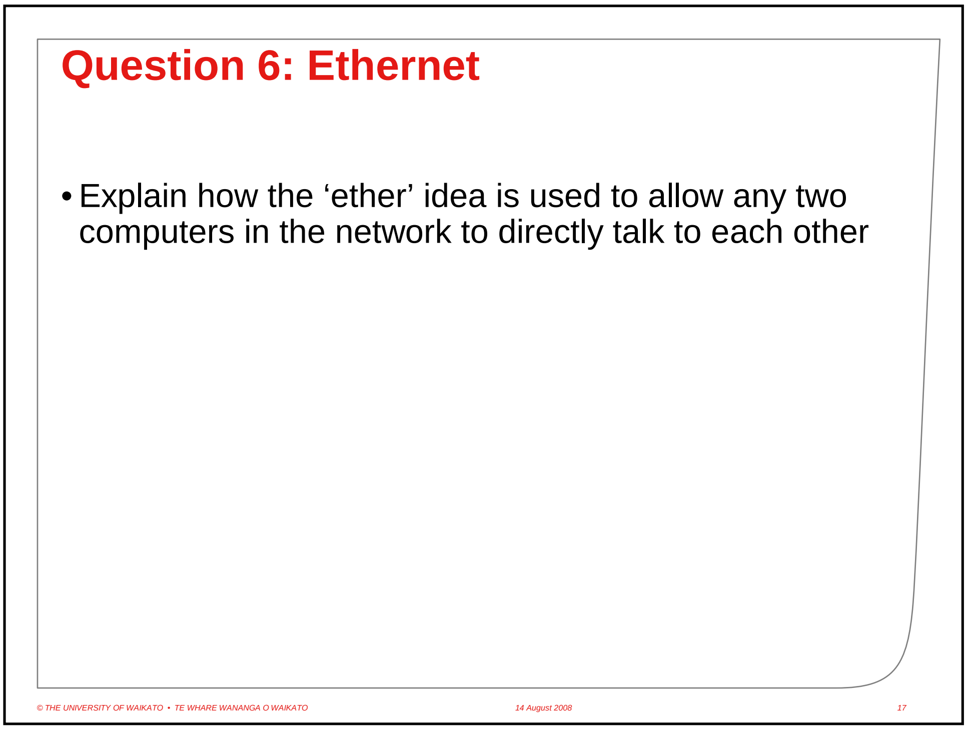
Question 5: Asynchronous I/O

- How does the receiver know when each bit in the message arrives?



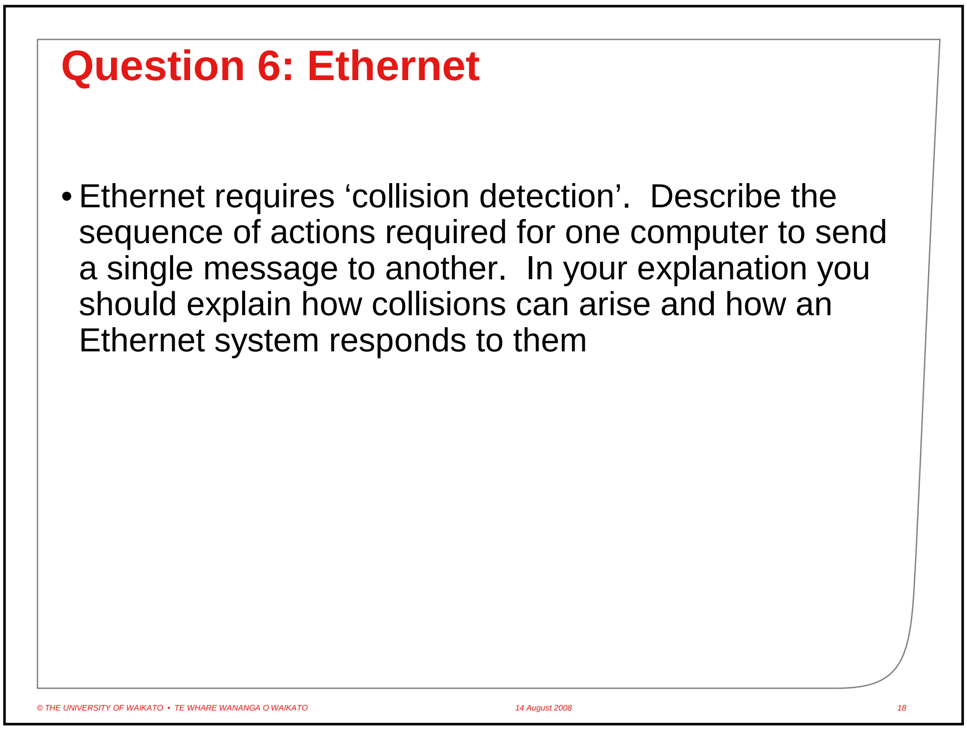
Question 5: Asynchronous I/O

- Why do small timing inaccuracies (differences between sender and receiver) not cause errors?



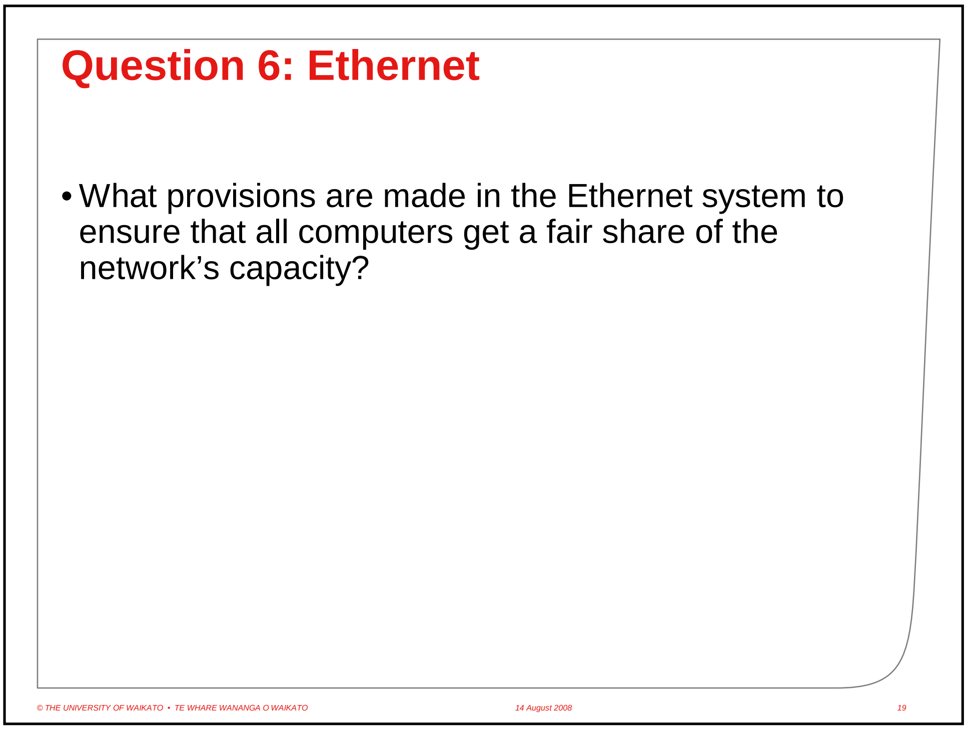
Question 6: Ethernet

- Explain how the ‘ether’ idea is used to allow any two computers in the network to directly talk to each other



Question 6: Ethernet

- Ethernet requires ‘collision detection’. Describe the sequence of actions required for one computer to send a single message to another. In your explanation you should explain how collisions can arise and how an Ethernet system responds to them



Question 6: Ethernet

- What provisions are made in the Ethernet system to ensure that all computers get a fair share of the network's capacity?