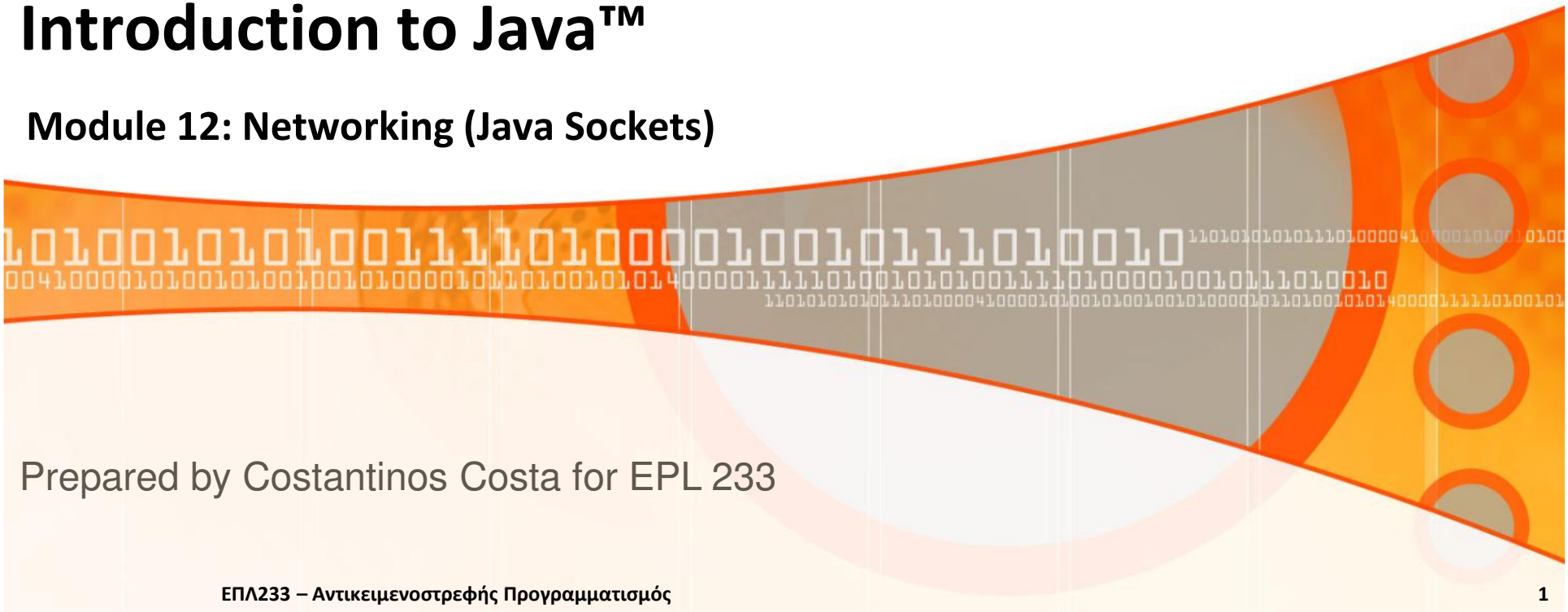
Introduction to Java™

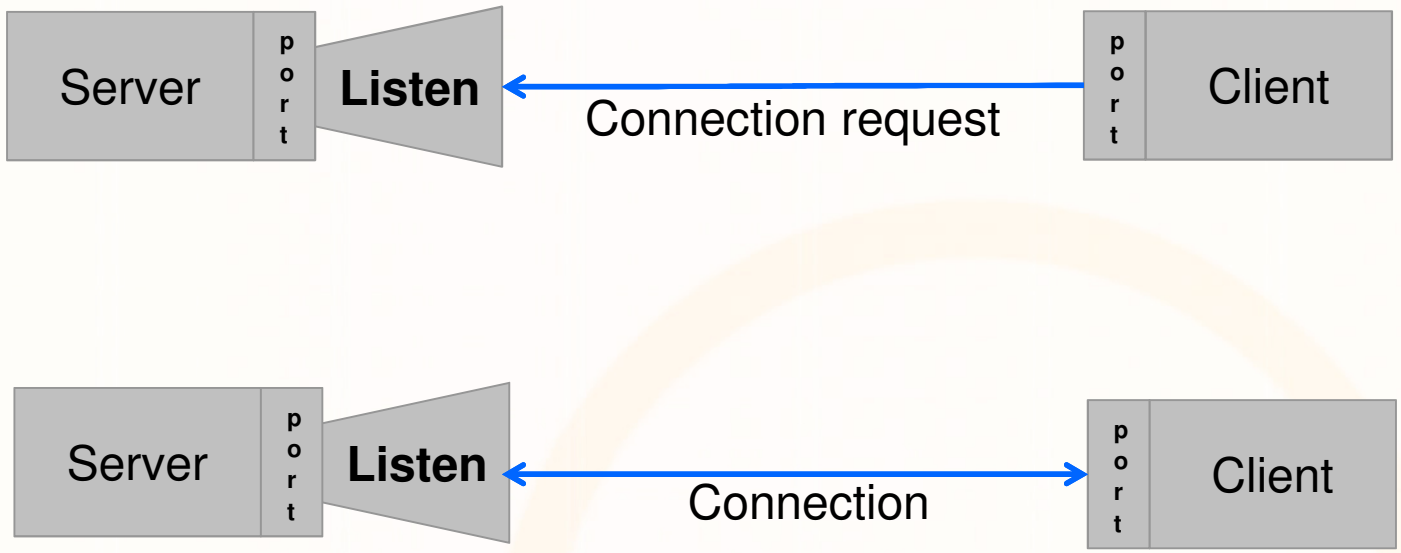
Module 12: Networking (Java Sockets)



Prepared by Costantinos Costa for EPL 233

What Is a Socket?

- A socket is one end-point of a two-way communication link between two programs running on the network. Socket classes are used to represent the connection between a client program and a server program. The java.net package provides two classes--Socket and ServerSocket--that implement the client side of the connection and the server side of the connection, respectively.



Reading from and Writing to a Socket in Java

- Let's look at a simple example that illustrates how a program can establish a connection to a server program using the Socket class and then, how the client can send data to and receive data from the server through the socket.So,
    - Download the code.
    - Run the code.
    - Observer the code.

EchoClient

```
package example;
import java.io.*;
import java.net.*;

public class EchoClient {
Socket echoSocket = null;
PrintWriter out = null;
BufferedReader in = null;

public EchoClient() {
// TODO Auto-generated constructor stub
try {
     // Create a new socket for the client
     echoSocket = new Socket("localhost", 8080);
     // OutputStream to send messages to the Server
     out = new PrintWriter(echoSocket.getOutputStream(), true);
     // InputStream to get messages from the Server
     in = new BufferedReader(new InputStreamReader(echoSocket.getInputStream()));
} catch (UnknownHostException e) {
     System.err.println("Don't know about host: localhost.");
     System.exit(1);
} catch (IOException e) {
     System.err.println("Couldn't get I/O for "
     + "the connection to: localhost.");
     System.exit(1);}
}
```

EchoClient

```
public void run() {
      // Create a buffer reader for the user input
      BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));
      String userInput = "";
      try {
      System.out.println(in.readLine());
            while (!userInput.equals("bye")) {
                  userInput = stdIn.readLine();
                  out.println(userInput);
                  System.out.println("echo: " + in.readLine());}
      } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
      }
      // Housekeeping
      try {
            out.close();
            in.close();
            stdIn.close();
            echoSocket.close();
      } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();}
      }
public static void main(String[] args) throws IOException {
      EchoClient echoClient = new EchoClient();
      echoClient.run();
      }
}
```

EchoServer

```
package example;

import java.io.*;
import java.net.*;

public class EchoServer {

private ServerSocket server;

public EchoServer(int portnum) {
    try {
        // Create new socket that listen any new requests
        server = new ServerSocket(portnum);
    } catch (Exception err) {
         System.out.println(err);
    }
}
```

EchoServer

```
public void run() {
     try {
     System.out.println("EchoServer is started!");
     // Accept is a blocking operation
     while (true) {
          // Get client socket
          Socket client = server.accept();
          BufferedReader r = new BufferedReader(new InputStreamReader(client.getInputStream()));
          PrintWriter w = new PrintWriter(client.getOutputStream(), true);
          w.println("Welcome to the Java EchoServer.  Type 'bye' to close.");
          String line;
          // Actual Communication with the client
          do {
               line = r.readLine();
               if (line != null)
               w.println(line);
          } while (!line.trim().equals("bye"));
          client.close();}
     } catch (Exception err) {
     System.err.println(err);
     }}
     public static void main(String[] args) {
          EchoServer s = new EchoServer(8080);
          s.run();
     }
}
```

Task

- Create multithreaded chat room based on the below protocol.
- Download the template code and follow the `TODO` comments **inside** the template code.