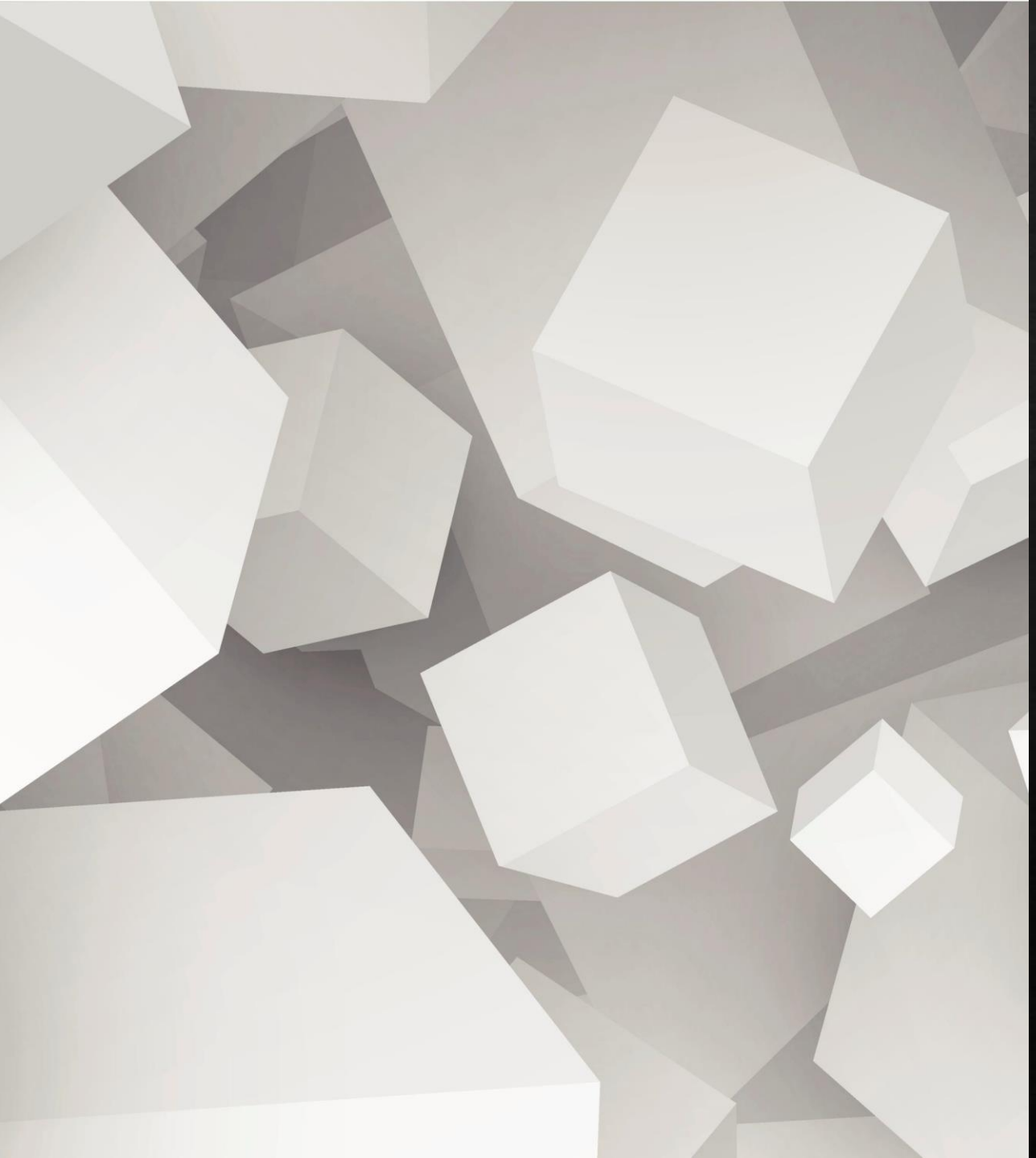
NETWORKING IN JAVA

MEHRSHAD SAADATINIA



Contents

- I/O streams
- Socket Programming Overview
- Serialization
- How to handle multiple clients?

I/O Streams

- DataInputStream & DataOutputStream : Binary read and write (readUTF & writeUTF)
- BufferedInputStream & BufferedOutputStream : uses buffer for read and write
- ObjectInputStream & ObjectOutputStream : serialization & deserialization
- FileInputStream & FileOutputStream : creates a stream from or to a file

Decorator design-pattern in IO Streams

```
FileOutputStream file = new FileOutputStream("c:/f.txt");
BufferedOutputStream buffer = new BufferedOutputStream(file);
PrintStream print = new PrintStream(buffer);
print.println("salam");
```

serialization

Sending Objects through a data stream is called serialization

Reconstruction of the object from data stream is called deserialization

This utility is possible for any class which implements Serializable:

```
public interface Serializable {}
```

Socket programming overview

Provides connection between 2 computers based on the Client – Server architecture

WHAT IF WE NEED TO HANDLE MULTIPLE CLIENTS ?

Thread methods we need

- `public static Thread currentThread()`: returns the currently executing thread
- `public long getId()`: returns this thread's ID
- `Thread.sleep(milliseconds)`: pauses the current thread for specified *milliseconds*