Java

Networking

InetAddress

- Java has a class java.net.InetAddress which abstracts network addresses.
- *Major methods*
  - *getLocalHost()*
  - *getByAddress()*
  - *getByName()*
- **Example :**
  - *HostInfo.java*
  - *AddressGenerator.java*
  - *Resolver.java*

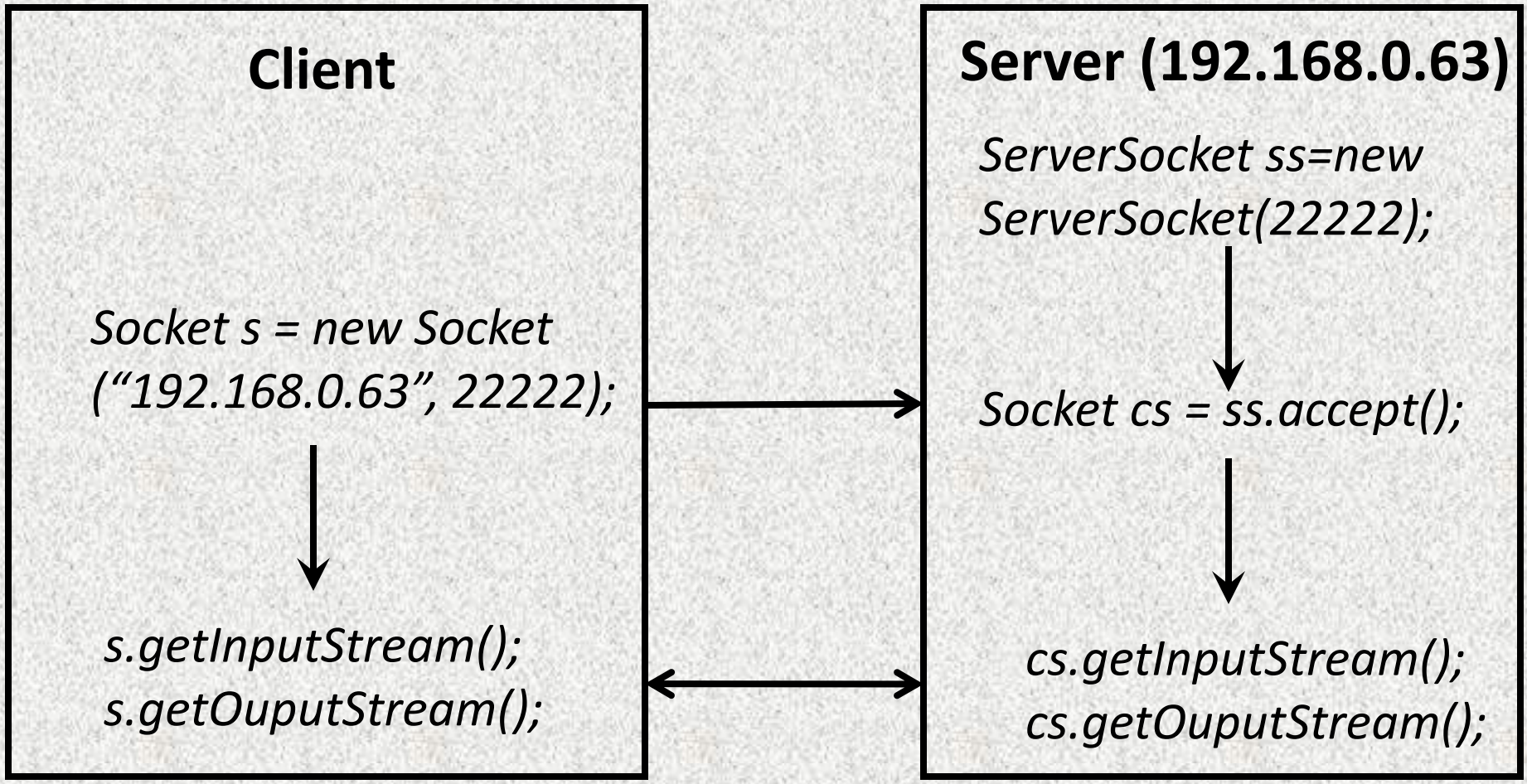
TCP

- TCP stands for Transmission Control Protocol.
- TCP is connection-oriented.
- It provides reliability.
- What is *Server* and *Client* ?
  - A server is a piece of software which advertises and then provides some service on request.
  - A client is a piece of software (usually on a different machine) which makes use of some service.

TCP Sockets

- Two types of TCP Sockets.
  - ***ServerSocket***
    - ServerSocket is used by servers so that they can accept incoming connections from client.
  - ***Socket***
    - Socket is used by clients who wish to establish a connection to a (remote) server.

Scenario


Client
Socket s = new Socket
("192.168.0.63", 22222);
s.getInputStream();
s.getOuputStream();
Server (192.168.0.63)
ServerSocket ss=new
ServerSocket(22222);
Socket cs = ss.accept();
cs.getInputStream();
cs.getOuputStream();

TCP Sockets

- ***Example:***
  - *Server.java*
  - *Client.java*
  - *ReadThread.java*
  - *WriteThread.java*
  - *NetworkUtil.java*
  - *Data.java*

UDP

- UDP stands for User Datagram Protocol.
- UDP is not connection-oriented.
- It does not provide reliability.
- It sends and receives packets known as Datagram.

Datagram Packet & Socket

- One type of Packet and one type of Socket.
  - ***DatagramPacket***
    - Used to encapsulate Datagram.
  - ***DatagramSocket***
    - DatagramSocket is used by both server and client to receive DatagramPacket.
- ***Example:***
  - *DatagramServer.java*
  - *DatagramClient.java*

End