Application Design Using Java

Lecture 10

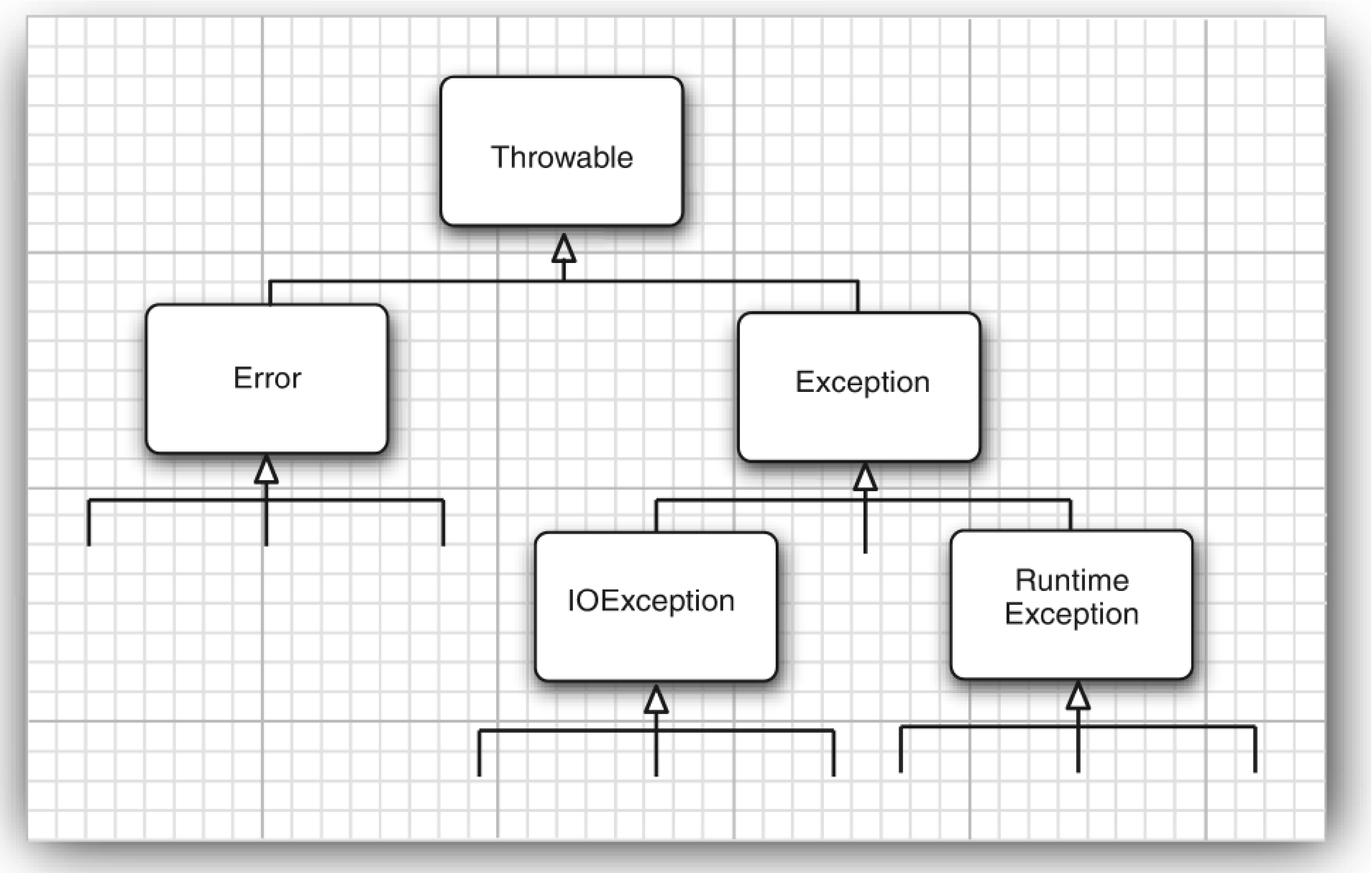
Assertions

- assert *condition* or assert *condition*: *expression*;
- AssertionError
- Disabled by default
  - java -enableassertions (java -ea)
  - java -disableassertions (java -da)
  - java -ea: AssertionsTest -ea:edu.rpi.csci4963.gol -da: edu.rpi.csci4963.gol.Board
  - -enablesystemassertions/-esa to enable assertions in system classes

Error Handling

- When an error is encountered
  - Recover to a safe state and enable the user to continue
  - Allow the user to save all work and terminate the program gracefully
- Types of errors
  - User input errors
  - Device errors
  - Physical limitations
  - Code errors
- Handling
  - Return a special “magic” error code
  - Exceptions

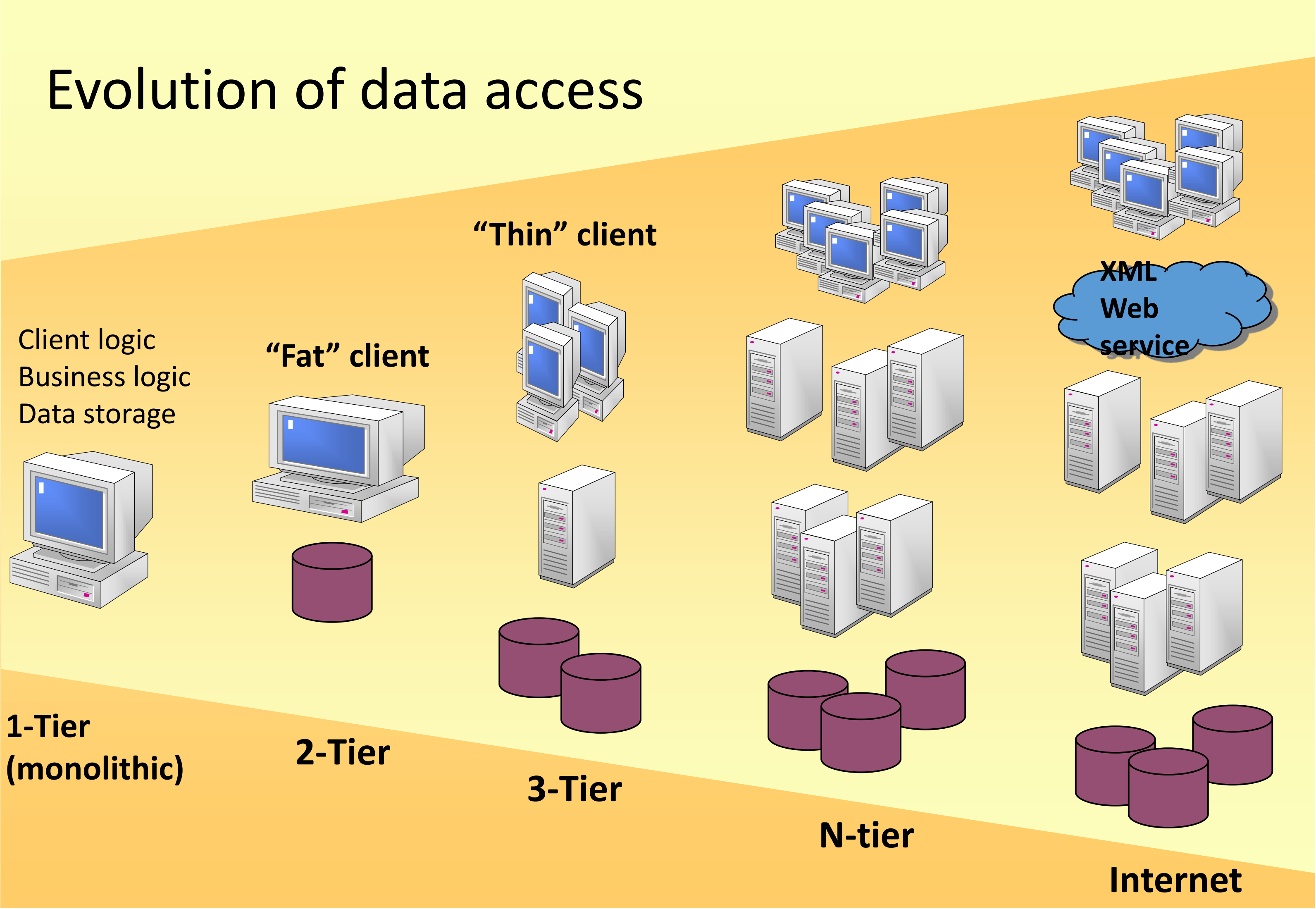
Exceptions


Throwable
Error
Exception
IOException
Runtime Exception

Exceptions

- Exception handling is not supposed to replace a simple test
- Do not micromanage exceptions
- Make good use of the exception hierarchy
- Do not squelch exceptions
- When you detect an error, “tough love” works better than indulgence
- Propagating exceptions is not a sign of shame

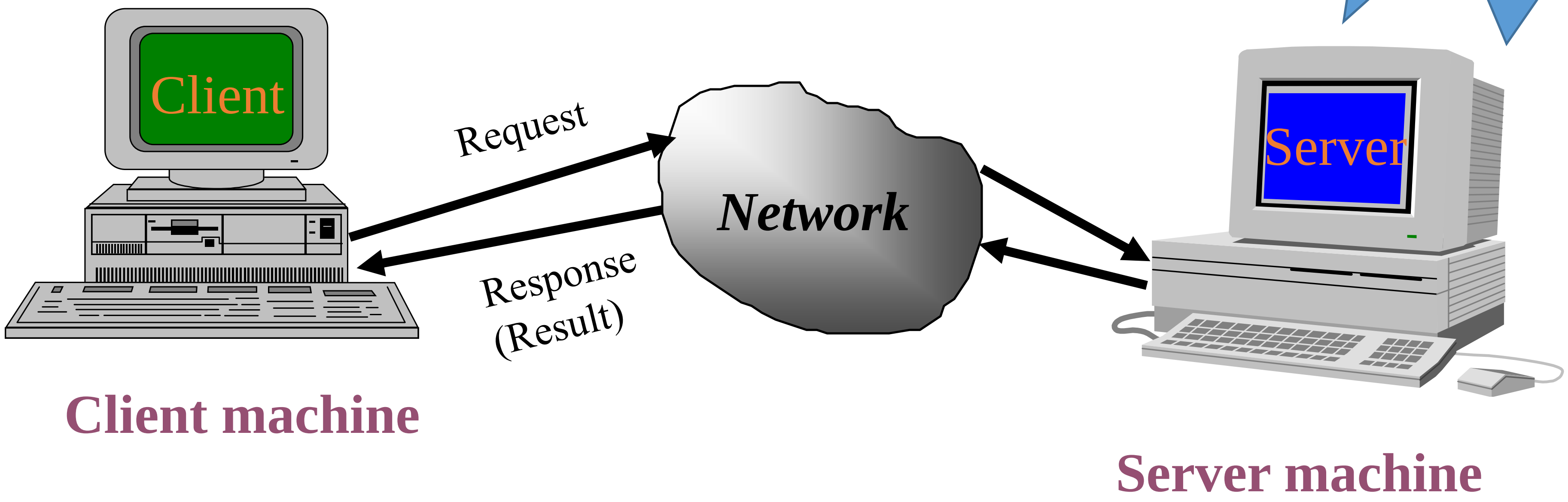
Data Access Application Models



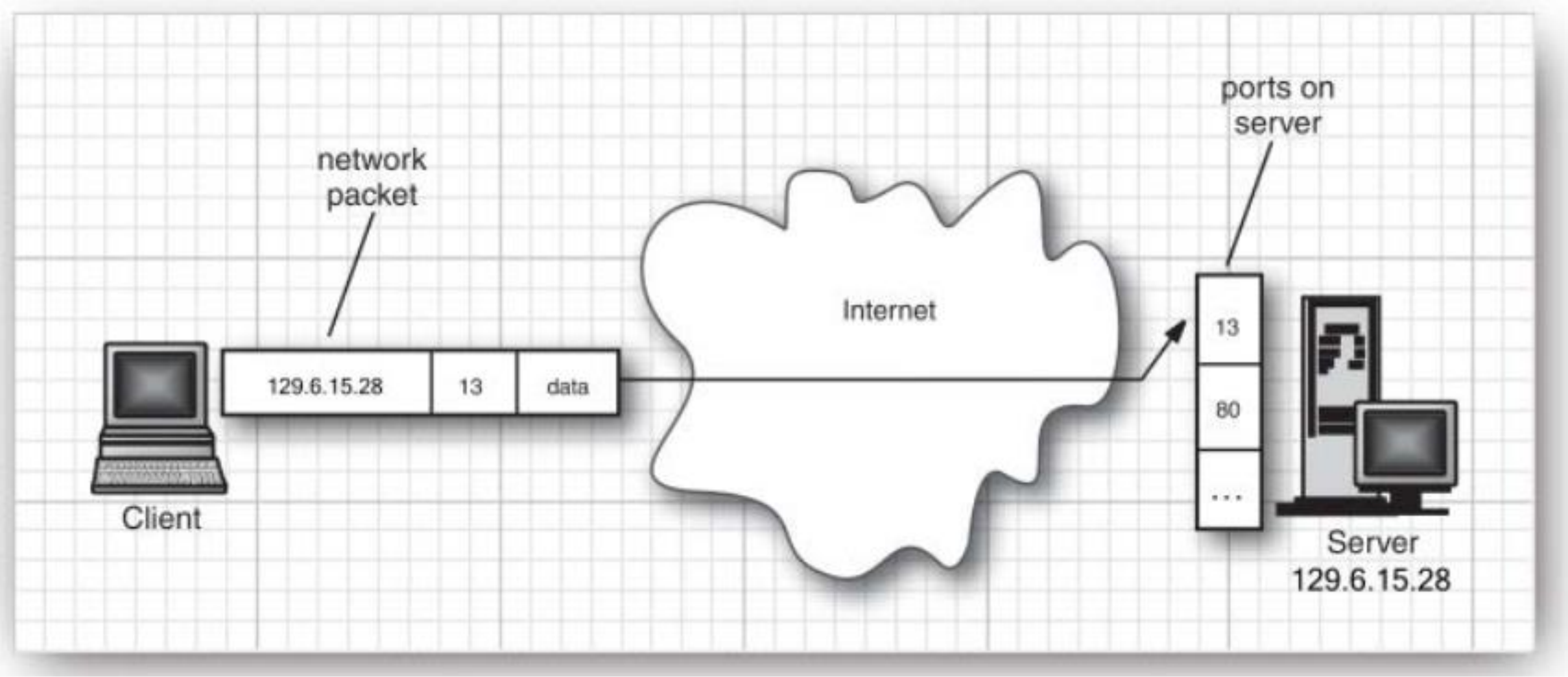
Networking

CSCI-4220 Network Programming

a client, a server, and network



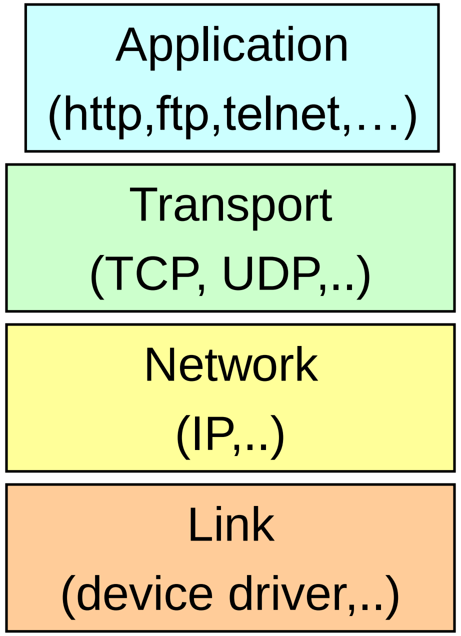
Client - server



Networking Basics

- Applications Layer
  - Standard apps
    - HTTP
    - FTP
    - Telnet
  - User apps
- Transport Layer
  - TCP
  - UDP
  - Programming Interface:
    - Sockets
- Network Layer
  - IP
- Link Layer
  - Device drivers

- TCP/IP Stack



Application
(http,ftp,telnet,…)

Transport
(TCP, UDP,..)

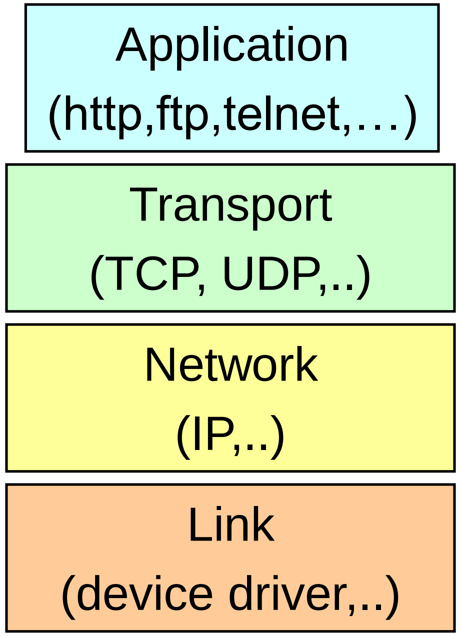
Network
(IP,..)

Link
(device driver,..)

Networking Basics

- TCP (Transport Control Protocol) is a connection-oriented protocol that provides a reliable flow of data between two computers.
- Example applications:
  - HTTP
  - FTP
  - Telnet

- TCP/IP Stack



Application
(http,ftp,telnet,…)

Transport
(TCP, UDP,..)

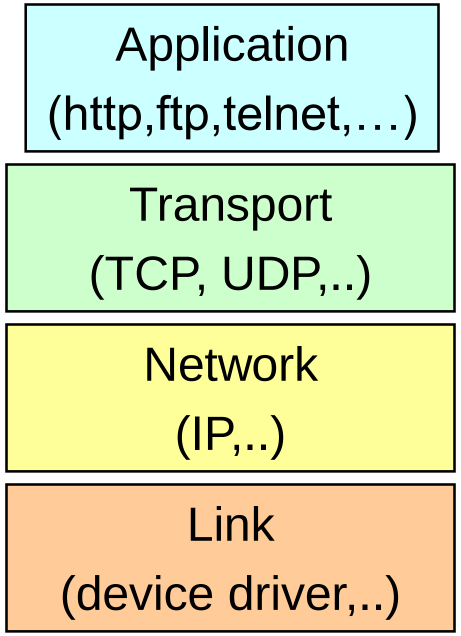
Network
(IP,..)

Link
(device driver,..)

Networking Basics

- UDP (User Datagram Protocol) is a protocol that sends independent packets of data, called *datagrams*, from one computer to another with no guarantees of delivery.
- Example applications:
  - Clock server
  - Ping

- TCP/IP Stack



Application

(http,ftp,telnet,…)

Transport

(TCP, UDP,..)

Network

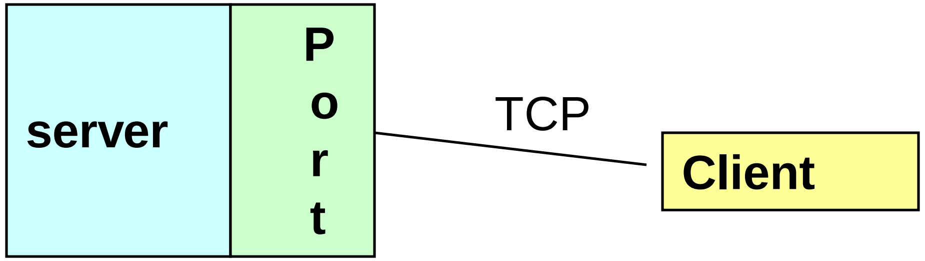
(IP,..)

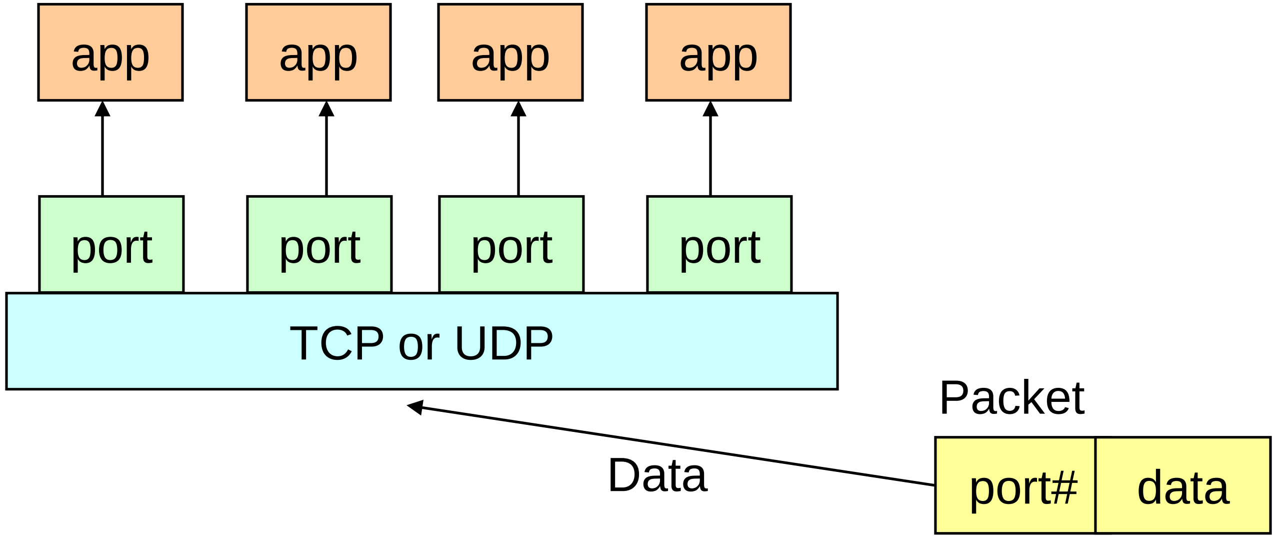
Link

(device driver,..)

Understanding Ports

- The TCP and UDP protocols use *ports* to map incoming data to a particular *process* running on a computer.

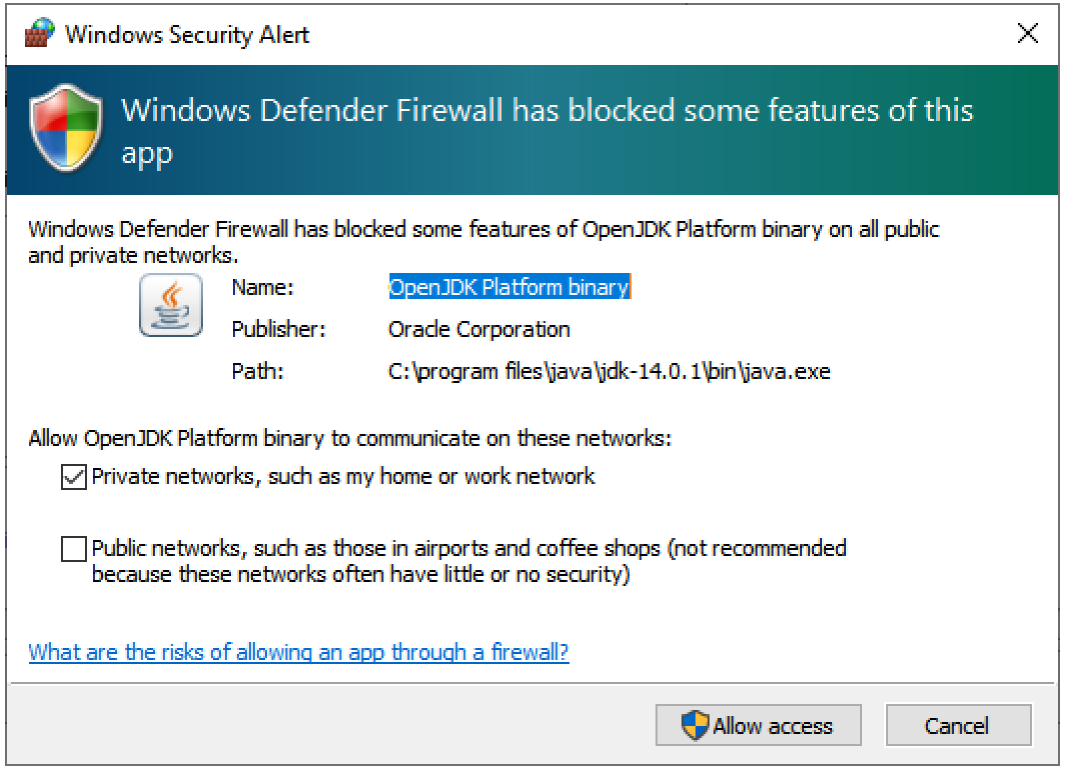




Understanding Ports

- Port is represented by a positive (16-bit) integer value
- Some ports have been reserved to support common/well known services:
  - ftp 21/tcp
  - telnet 23/tcp
  - smtp 25/tcp
  - login 513/tcp
- User level process/services generally use port number value >= 1024

Firewalls



Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of OpenJDK Platform binary on all public and private networks.

Name: OpenJDK Platform binary

Publisher: Oracle Corporation

Path: C:\program files\java\jdk-14.0.1\bin\java.exe

Allow OpenJDK Platform binary to communicate on these networks:

☑ Private networks, such as my home or work network

□ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

//TODO before next lecture:

- Practice problems
- Homework 2 due on 3/4 at 11:59 pm EST. Must be submitted on Submitty.
- Java puzzler (posted on Submitty Forum)