Week 10

* Message oriented communication
  + Transient messaging
    - Sockets
    - Message passing interface
* Socket Interface Operations
* Message passing interface – MPI
  + Standard developed to provide message passing operations with synchronous and asynchronous variants
  + MPI is designed for parallel applications (use transient communication)
  + Assumes communication takes place within a known group of processes (Comm World, groupID)
  + Applications use the MPI interface via a message passing library (C, C++, Fortran)
  + Example of MPI point to point communication
* Message Queuing System

Thursday

* What is RMI
  + RMI is a Java-specific object oriented extension of the Remote Procedure Call
  + It allows an object in one Java Virtual Machine to interact with the object in another
* Overview of RMI
  + RMI apps comprimse two separate programs, a server and client
  + The Server
    - Creates some remote objects
    - Makes references to these objects accessible
    - Waits for clients to invoke methods on these
  + The client
    - Obtains a remote reference to one or more remote objects on a server
    - Inokes methods on them
  + To send a message to a remote server
    - The client object has to find the object
      * Looks it up in a registry
    - The client object then has to marshal the parameters
      * Java requires serializable parameters
      * The server object has to unmarchal its parameters, do its computation
        + Then marshal the response
      * The client object has to unmarchal the response
* Terminology
  + Remote object
    - An object on another computer
  + Client object
    - The object making the request
  + Server object
    - The object receiving the request
  + Rmiregistry
    - A special server that looks up objects by name
  + Rmic
    - A special compiler for creating stub and skeleton classes
* RMI components
  + Server hosting a remote object
    - Construct an implementation of the object
    - Provide access to methods via skeleton
    - Register (bind) the object to the RMI registry
  + RMI registry
    - Each remote object needs to register their location
    - RMI clients find remote objects via the lookup service
  + Client using a remote object
    - Ask registry for location of the object (lookup)
    - Construct stub
    - Call methods via the objects stub
* RMI architecture
  + Transparency regarding distribution
    - RMI offers full transparency
  + RMI client and RMI server
    - Both client and server are normal objects implemented in java
    - Server documents the interface it provides for remote access
  + Stub and Skeleton
    - Additional classes automatically created by a special compiler
    - Take care of communication handling between client and server
  + Stub (client)
    - Responsible for sending the remote call over to the server side skeleton
    - Marshaling the object parameters
  + Skeleton
    - Takes the calls of the stub, unmarshals the parameters
    - Forwards the call to the server object
    - Waits for the result
    - Sends the result back to the stub
* Java Interfaces and Classes
  + Interfaces define behavior
  + Classes define implementation
  + Therefore
    - In order to use a remote object, the client must know its behaviour (interface), but does NOT need to know its implementation
    - In order to provide an object, the