Java Examples

Simple Program Example
- Much Syntax similar to C
  - // Import statements are used in include libraries
    - import java.io.*;
  - // By default a class will extend the object class, but we
    - // could make it extend any class.
  - // The implements keyword specifies that a class will
    - implement a Java Abstract Interface.
  - public class demo extends Object implements Runnable
    - {
  - import java.io.*;
    - public demo()
      - {
        - // This is the closest thing to a constant in Java.
        - private static final int CONSTANT = 0;
      - }
  - private static final int CONSTANT = 0;
  - private short MyArray[][][];
  - public demo()
    - {
      - // Empty Constructor
        - {
        - public demo(int length)
          - {
            - nLength = length;
        - }
  - / /  C l a s s  w i d e  v a r i a b l e s  c a n  b e  p u b l i c  o r  p r i v a t e
    - private int nLength;
  - private short MyArray[][];
  - / /  I n c l u d i n g  t h i s  f u n c t i o n  m a k e s  a n  c l a s s  e x e c u t a b l e !
    - public static void main(String[] args)
      - {
        - // There are many subclasses of the
        - Exception class which describe many
        - specific error types.
        - catch (Exception e)
          - {
            - System.out.println("Error: " +
            - e.toString()));
        - }
    - }
    - }
Java Examples

Event Listeners

Java uses Listeners to listen for and trap GUI events.
- When an event occurs the listener either handles the
  event or invokes an event handler routine.
  - ActionListener
    - Listens for DO events, like clicking a mouse button.
  - ChangeListener
    - Listens for state change events.
  - ComponentListener
    - Listens for component moved, resized, hidden, shown
      events.
Simple Applet Example

- Applet's run in web browsers.
- Best to stay away from SWING objects unless you want to break web browser compatibility or use Swing browser plugins.

Multithreading

- (2) ways to implement multithreaded apps in Java
  - Extend the Thread class.
    - Must include public void run() method.
    - Public class MyServer extends Thread
    - MyServer srv = new MyServer();
    - MyServer.start();
  - Implement the Runnable abstract interface.
    - Must include public void run() method.
    - Thread t = new Thread(this);
    - t.start();

Multithreading

- Implementing the Runnable abstract interface.
  - Defines work for the thread to do.
  - A Class can instantiate a Thread instance and then pass itself in as a reference to make a new Thread associated with itself.
  - This is important because classes should not be subclassed unless the programmer intends on modifying or enhancing the fundamental behavior of the class.

Resizing Example

- Any bitmap can be resized.
- Graphics class defines graphics methods.
- drawImage() method offers ability to specify width/height dimensions.

Persistent Background

- Keep a copy of the persistent background image in memory. (invisible)
- Render your current image to a memory image. (invisible)
- Start by copying the persistent background image to the rendering image in memory. (invisible)
- Perform renderings on memory image
- Paint rendering memory image to display.

Persistent Background

- Perform only 1 paint to the screen
- All other paints are invisible to user
- Less Flicker
- Use Threads!
- Paint method can be overridden to automatically paint & render graphics.
- Be sure not to override painting of desirable Java awt/Swing controls however.
- Does anyone have a better idea?
Socket Client Application

- AWT Applet
- Runs in web browser
- Can only connect to IP address of web server. Any port number.

Socket Client Application

- Socket Class
  - Socket.getInputStream()
  - Socket.getOutputStream()
- Use a separate Thread to process input data.
  - Input Stream methods block until there is data.
  - Can you use event invocation to handle incoming data from Socket?

Socket Server Application

- Stand alone Java program
- Multithreaded / multi-connection
- ServerSocket() class listens for and accepts connections.
- When connection is made a Socket() class is returned.
- Socket class instance is handed off to thread to implement the conversation.

Java RMI

- Remote Method Invocation
- Remote Java Classes
- Server Side Classes
  - Implement system/business logic
  - No knowledge of remote client or RMI server.
  - Just a dumb class.
- RMI Server
  - Provides Implementation for Abstract Interface (shared by remote client)
  - Call Server Side Classes to implement functionality

Java RMI

- RMI Client
  - Uses same Abstract Interface file to know prototypes of functions to call.
- Rmic skeleton/stub compiler.
- RMI Client/Server must have same skeleton/stub files.
- Rmi Registry stores

Java RMI

- Function Callbacks
  - Can be done remotely (across machines using RMI)
  - Can also be done locally (Not just RMI)
  - Must pass an interface reference to the caller.
  - A function callback invokes the client program to respond.
  - Better than polling until a condition is met.
Java RMI Example