Solutions to ITEC 1630 December 20, 2006 Final Exam

Part A

1a 2a 3a 4b 5b 6a 7a 8b 9a 10a 11f 12a 13b 14a 15b 16a 17d 18c

Part B

B1.

--- Add public synchronized boolean isEmpty() { return firstNode == null; }

```
public synchronized void insertAtFront( Object insertItem ) {
    if ( isEmpty() )
      firstNode = lastNode = new ListNode( insertItem );
    else
      firstNode = new ListNode( insertItem, firstNode );
  }
```

B2.

89 28 42 30 72 22 18 6 44 8 36 14 55 67 21

Part C

```
import java.io.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class ReadRandFile {

  RandomAccessFile input;
  Record data;
  Record[] emp;

  public ReadRandFile() {
    data = new Record();
    int count=0;

    try {
      input = new RandomAccessFile( "Univ.dat", "r" );
      emp = new Record[input.length() / data.size() ]
       while (input.getFilePointer() < input.length()){
       if ( data.getID() != 0 )
           emp[count++] = data;
       }
    }
    catch ( IOException e ) {
      System.err.println( e.toString() );
      System.exit( 1 );
    }

    bubbleSort(emp, count);

    for (int i=0; i<count; i++)
       System.out.println(emp.toString())
  }

 public void bubbleSort(Record e[], int c){
   for (int pass =1; pass<c; pass++)
     for(int j=0; j<c-1; j++)
       if(ej].getYears()<e[j+1].getYears()) {
         Record hold=e[j];
         e[j]=e[j+1];
         e[j+1]=e[j];
       }
  }

  public static void main( String args[] )   {
    ReadRandFile accounts = new ReadRandFile();
  }
}
```

Part D

```
import java.awt.event.*;
import java.io.*;
import java.net.*;
import java.awt.*;
import javax.swing.*;

public class Client3 extends JApplet implements ActionListener{
            Socket s = null;
            boolean good = false;
            DataInputStream dis = null;
            DataOutputStream dos = null;
            JTextField t1, t2, t3;
            JButton send;
            JTextArea ta;

            public void init() {
             Container c = getContentPane();
              c.setLayout(new BorderLayout());
               c.add(new JLabel("Welcome to KMart"), BorderLayout.NORTH);
               JPanel p1 = JPanel();
               p1.setLayout (new GridLayout(4,2);
               p1.add(new JLabel(“Client Number”));
                t1 = new JTextField(10);
                 p1.add(t1);
               p1.add(new JLabel(“Item Name”));
                t2 = new JTextField(10);
                 p1.add(t2);
               p1.add(new JLabel(“Number of Items”));
                t3 = new JTextField(10);
                 p1.add(t3);
               send = new JButton("Send");
               send.addActionListener(this);
              p1.add(butt);
```

```
      c.add(p1, BorderLayout.CENTRE);
      ta = new JTextArea(;
      c.add(ta, BorderLayout.SOUTH);
    }

    public void start() {
      word.setText("");
      try {
           s = new Socket("www.kmart.com", 8453);
           dis = new DataInputStream(s.getInputStream());
           dos = new DataOutputStream(s.getOutputStream());
           def.setText("(connected to server)");
           good = true;
      } catch (IOException e) {
           def.setText("Cannot create Socket");
      }
    }

    public void stop() {
      if (good)
           try {
             s.close();
           }
           catch (IOException e) {
           }
      s = null;
      dis = null;
      dos = null;
      good = false;
    }

    public void actionPerformed(ActionEvent e) {
             String sin = t2.getText();
             String sout = null;
             if (good) {
                  try {
                    dos.writeUTF(sin);
                    dos.flush();
                    sout = dis.readUTF();
                  }
                  catch (IOException ex) {
                    sout = "(Socket died)";
                    good = false;
                  }
             } // end of if (good)

             double price = Double.parseDouble(sout);
             int nr=Integer.parseInt(t3.getText()) ;
             String s = « Client Number « + t1.getText()+ “/n “+ sin +
                 “ total price= “ + price*nr;
             ta.setText(s);
             t1.setText(“”);
             t2.setText(“”);
             t3.setText(“”);
    }
}
```

```
import java.net.*;
import java.io.*;


class Sockthread extends Thread {
        Socket s = null;

        Sockthread(Socket sock) {
          s = sock;
          start();
        }

        public void run() {
          try {
                DataInputStream dis = new DataInputStream(s.getInputStream());
                DataOutputStream dos =
                 new DataOutputStream(s.getOutputStream());
                boolean keepgoing = true;
                while (keepgoing) {
                   String str = dis.readUTF();
                   if (str.equals(""))
                         keepgoing = false;
                   else {
                         dos.writeUTF(database(str));
                         dos.flush();
                   }
                }
                s.close();
                System.out.println("Client terminated");
          } catch (IOException e) {
                System.out.println("Client aborted prematurely");
          }
        }
}

public class Server {
        public static void main(String args[]) {
          ServerSocket ss = null;
          boolean good = true;
          try {
                ss = new ServerSocket(21346, 100);
          } catch (IOException e) {
                System.out.println("Cannot create server");
                good = false;
          }
          while (good) {
                Socket s = null;
                try {
                        s = ss.accept();
                        new Sockthread(s);
                        System.out.println("Starting new client from " +
                         s.getInetAddress());
                } catch (IOException e) {
                        System.out.println("Fatal server error number 78231");
                        good = false;
                    }
                }
            }
    }
```