1. [25 Pts] Write a program called JTableDemo that does the followings:

• Combine the TablePropertiesDemo and the TestTableSortFilter (sample programs shown in the class) functionalities together into one program

• The GUI should show the JTable at the top (with the same contents as the examples shown in class)

• Has the JSpinner to change the Row Height property and the CheckBox to set whether to show the grids or not (no need to add the JSpinner to set the Row Margins)

• Has the Sorter to sort the table when the column header is clicked (no need to add the filter TextField and the Filter button)

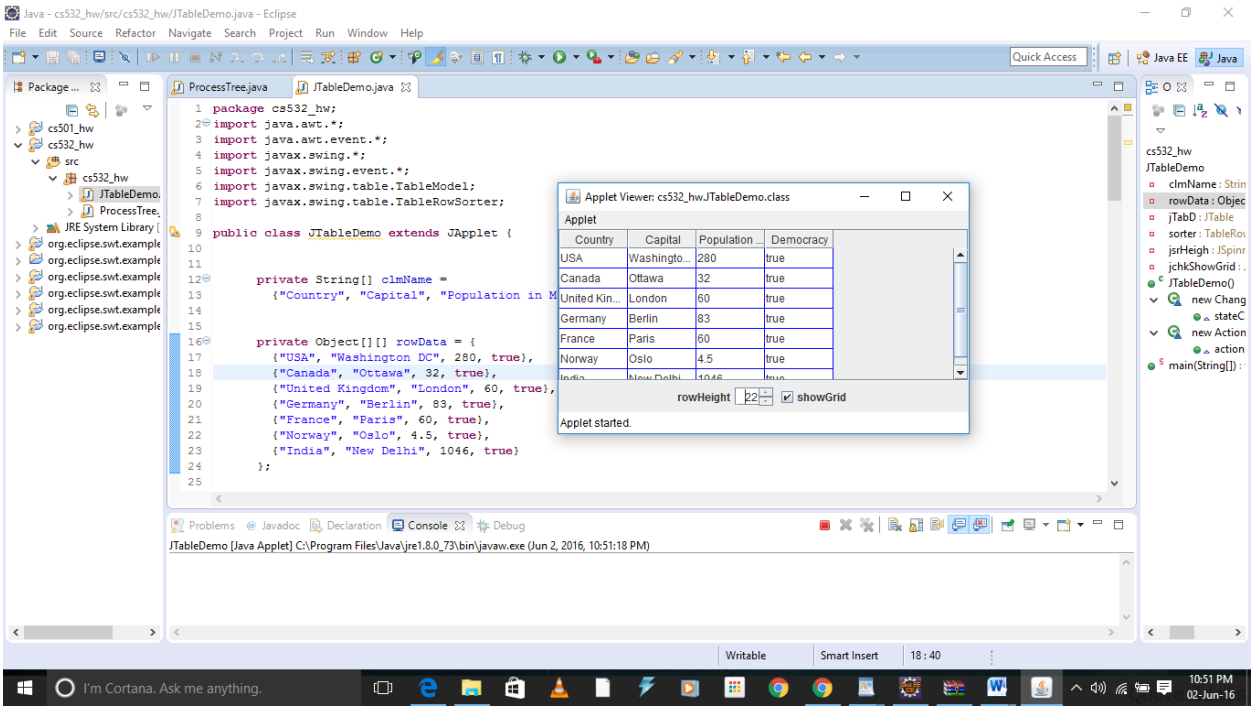
• See the images below for reference on how the application should look like and how it should work ;

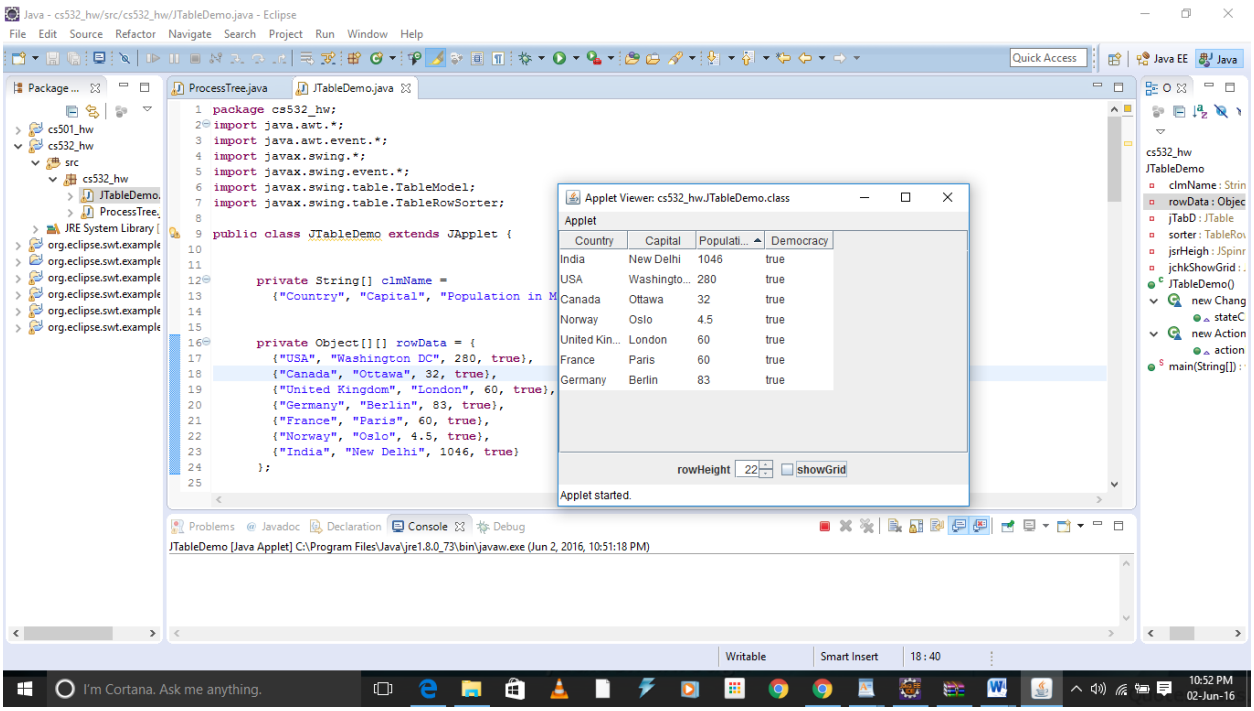
Output:

```
package cs532_hw;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

public class JTableDemo extends JApplet {


        private String[] clmName =
          {"Country", "Capital", "Population in Millions", "Democracy"};


        private Object[][] rowData = {
          {"USA", "Washington DC", 280, true},
          {"Canada", "Ottawa", 32, true},
          {"United Kingdom", "London", 60, true},
          {"Germany", "Berlin", 83, true},
          {"France", "Paris", 60, true},
          {"Norway", "Oslo", 4.5, true},
          {"India", "New Delhi", 1046, true}
        };

        // Create a table
        private JTable jTabD = new JTable(rowData, clmName);
        private TableRowSorter<TableModel> sorter =
            new TableRowSorter<TableModel>(jTabD.getModel());
```

```
// Create two spinners
private JSpinner jsrHeigh =
  new JSpinner(new SpinnerNumberModel(16, 1, 50, 1));


// Create a checkbox
private JCheckBox jchkShowGrid = new JCheckBox("showGrid", true);



public JTableDemo() {
  JPanel panel1 = new JPanel();
  panel1.add(new JLabel("rowHeight"));
  panel1.add(jsrHeigh);

  panel1.add(jchkShowGrid);

  jTabD.setRowSorter(sorter);



  add(panel1, BorderLayout.SOUTH);
 // add(panel2, BorderLayout.NORTH);
  add(new JScrollPane(jTabD));

  // Initialize jTabD
  jTabD.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
  jTabD.setGridColor(Color.BLUE);
  jTabD.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
  jTabD.setSelectionBackground(Color.RED);
  jTabD.setSelectionForeground(Color.WHITE);

  // Register and create a listener for jsrHeigh
  jsrHeigh.addChangeListener(new ChangeListener() {
    public void stateChanged(ChangeEvent e) {
      jTabD.setRowHeight(
        ((Integer)(jsrHeigh.getValue())).intValue());
    }
  });




  // Register and create a listener for jchkShowGrid
  jchkShowGrid.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
      jTabD.setShowGrid(jchkShowGrid.isSelected());
    }
  });

  // Register and create a listener for jcboAutoResizeMode

}

public static void main(String[] args) {
  JTableDemo applet = new JTableDemo();
```

```
            JFrame frame = new JFrame();
            //EXIT_ON_CLOSE == 3
            frame.setDefaultCloseOperation(3);
            frame.setTitle("JTableDemo");
            frame.getContentPane().add(applet, BorderLayout.CENTER);
            applet.init();
            applet.start();
            frame.setSize(400,320);
            Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
            frame.setLocation((d.width - frame.getSize().width) / 2,
                              (d.height - frame.getSize().height) / 2);
            frame.setVisible(true);
        }

}
```





2. [25 Pts] JTree Customization:

• Set-up so that you can run the ProcessTree program (included in the demo programs zip file) in your Eclipse IDE environment

• Modify the program so that it only shows the JTree on the left side (i.e., only shows 1 JTree)

• Remove the selection mode label and combo box, so that only the “Add Node” and the “Remove Selected Node” buttons are at the top (above the JTree)

• Add a Text Area below the JTree that displays a message when a tree node is selected, saying “Node [node’s label] is selected”

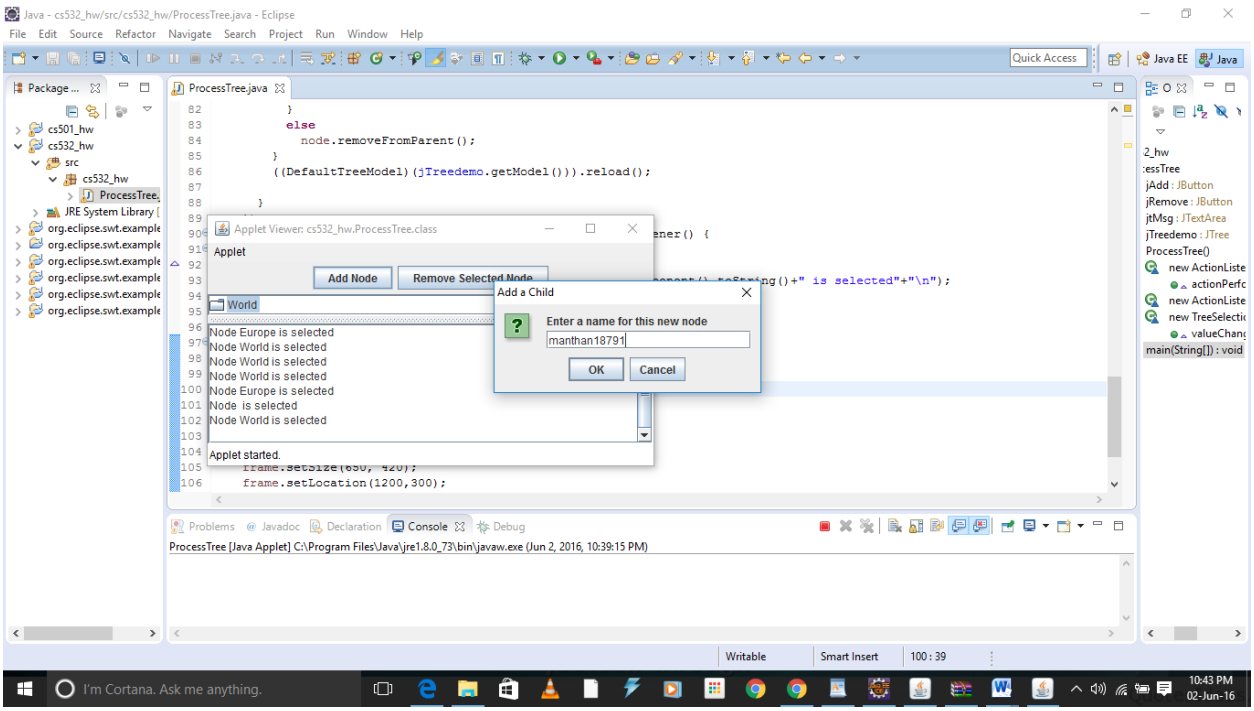
• See the screen capture below as an example of how the application should look like and work

OP:

```
package cs532_hw;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.*;
```

```
public class ProcessTree extends JApplet{

  private JButton jAdd = new JButton("Add Node");
  private JButton jRemove = new JButton("Remove Selected Node");
  private JTextArea jtMsg = new JTextArea();
  private JTree jTreedemo;

  public ProcessTree() {

    DefaultMutableTreeNode root, europe, northAmerica, us;

    europe = new DefaultMutableTreeNode("Europe");
    europe.add(new DefaultMutableTreeNode("UK"));
    europe.add(new DefaultMutableTreeNode("Germany"));
    europe.add(new DefaultMutableTreeNode("France"));
    europe.add(new DefaultMutableTreeNode("Norway"));

    northAmerica = new DefaultMutableTreeNode("North America");
    us = new DefaultMutableTreeNode("US");
    us.add(new DefaultMutableTreeNode("California"));
    us.add(new DefaultMutableTreeNode("Texas"));
    us.add(new DefaultMutableTreeNode("New York"));
    us.add(new DefaultMutableTreeNode("Florida"));
    us.add(new DefaultMutableTreeNode("Illinois"));
    northAmerica.add(us);
    northAmerica.add(new DefaultMutableTreeNode("Canada"));

    root = new DefaultMutableTreeNode("World");
    root.add(europe);
    root.add(northAmerica);


    JPanel p1 = new JPanel();
    p1.add(jAdd);
    p1.add(jRemove);
    JPanel p2 = new JPanel();
    p2.setLayout(new GridLayout(1,2));
    p2.add(new JScrollPane(jTreedemo = new JTree(root)));
    jtMsg.setWrapStyleWord(true);
    jtMsg.setLineWrap(true);
    getContentPane().add(p1, BorderLayout.NORTH);
    getContentPane().add(p2,BorderLayout.CENTER);
    getContentPane().add(new JSplitPane(JSplitPane.VERTICAL_SPLIT,p2,new
JScrollPane(jtMsg)), BorderLayout.CENTER);



    jAdd.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        DefaultMutableTreeNode parent =
(DefaultMutableTreeNode)jTreedemo.getLastSelectedPathComponent();
        if (parent == null) {
          JOptionPane.showMessageDialog(null,"No node in the first tree is
selected");
          return;
        }
```

```
        String nodeName = JOptionPane.showInputDialog(null, "Enter a name for
this new node", "Add a Child",JOptionPane.QUESTION_MESSAGE);
        parent.add(new DefaultMutableTreeNode(nodeName));
        ((DefaultTreeModel)(jTreedemo.getModel())).reload();
      }
    });

    jRemove.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        TreePath[] paths = jTreedemo.getSelectionPaths();
        if (paths == null) {
          JOptionPane.showMessageDialog(null,"No node in the left tree is
selected");
          return;
        }
        for (int i = 0; i < paths.length; i++) {
          DefaultMutableTreeNode node = (DefaultMutableTreeNode)
              (paths[i].getLastPathComponent());

          if (node.isRoot()) {
            JOptionPane.showMessageDialog(null,"Cannot remove the root");
          }
          else
            node.removeFromParent();
        }
        ((DefaultTreeModel)(jTreedemo.getModel())).reload();

      }
    });
    jTreedemo.addTreeSelectionListener(new TreeSelectionListener() {
        @Override
        public void valueChanged(TreeSelectionEvent e) {
             jtMsg.append("Node
"+e.getPath().getLastPathComponent().toString()+" is selected"+"\n");
        }
    });
  }
  public static void main(String[] args) {
    ProcessTree applet = new ProcessTree();
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(3);
    frame.setTitle("Tree Event Demo");
    frame.getContentPane().add(applet, BorderLayout.CENTER);
    applet.init();
    applet.start();
    frame.setSize(650, 420);
    frame.setLocation(1200,300);
    frame.setVisible(true);
  }


}
```


Java - cs532_hw/src/cs532_hw/ProcessTree.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Java
Package ...
cs501_hw
cs532_hw
src
cs532_hw
ProcessTree.
JRE System Library [
org.eclipse.swt.example
org.eclipse.swt.example
org.eclipse.swt.example
org.eclipse.swt.example
org.eclipse.swt.example
org.eclipse.swt.example
ProcessTree.java
82 }
83 else
84 node.removeFromParent();
85 }
86 ((DefaultTreeModel)(jTreedemo.getModel())).reload();
87
88 }
Applet Viewer: cs532_hw.ProcessTree.class
Applet
Add Node
Remove Selected Node
World
Node Europe is selected
Node World is selected
Node World is selected
Node World is selected
Node Europe is selected
Node is selected
Node World is selected
Applet started.
Add a Child
Enter a name for this new node
manthan18791
OK
Cancel
ener() {
ring()+" is selected"+"\n");
frame.setSize(650, 420);
frame.setLocation(1200,300);
2_hw
:essTree
jAdd : JButton
jRemove : JButton
jtMsg : JTextArea
jTreedemo : JTree
ProcessTree()
new ActionListe
actionPerfc
new ActionListe
new TreeSelectic
valueChang
main(String[]) : void
Problems
Javadoc
Declaration
Console
Debug
ProcessTree [Java Applet] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (Jun 2, 2016, 10:39:15 PM)
Writable
Smart Insert
100 : 39
I'm Cortana. Ask me anything.
10:43 PM
02-Jun-16