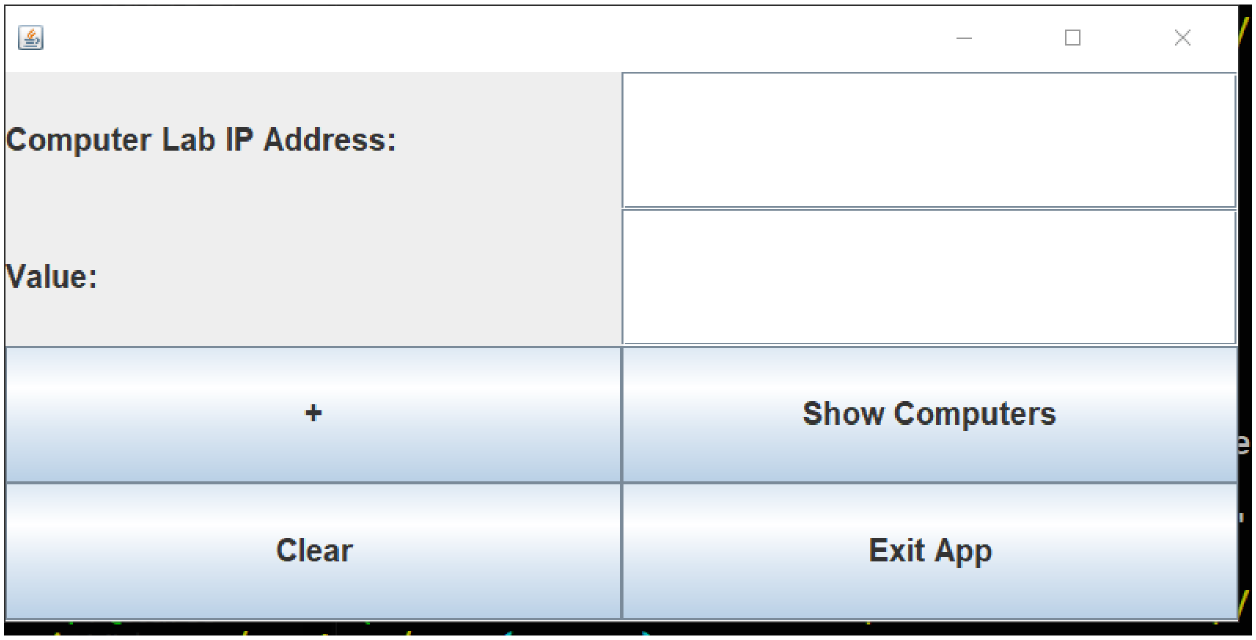
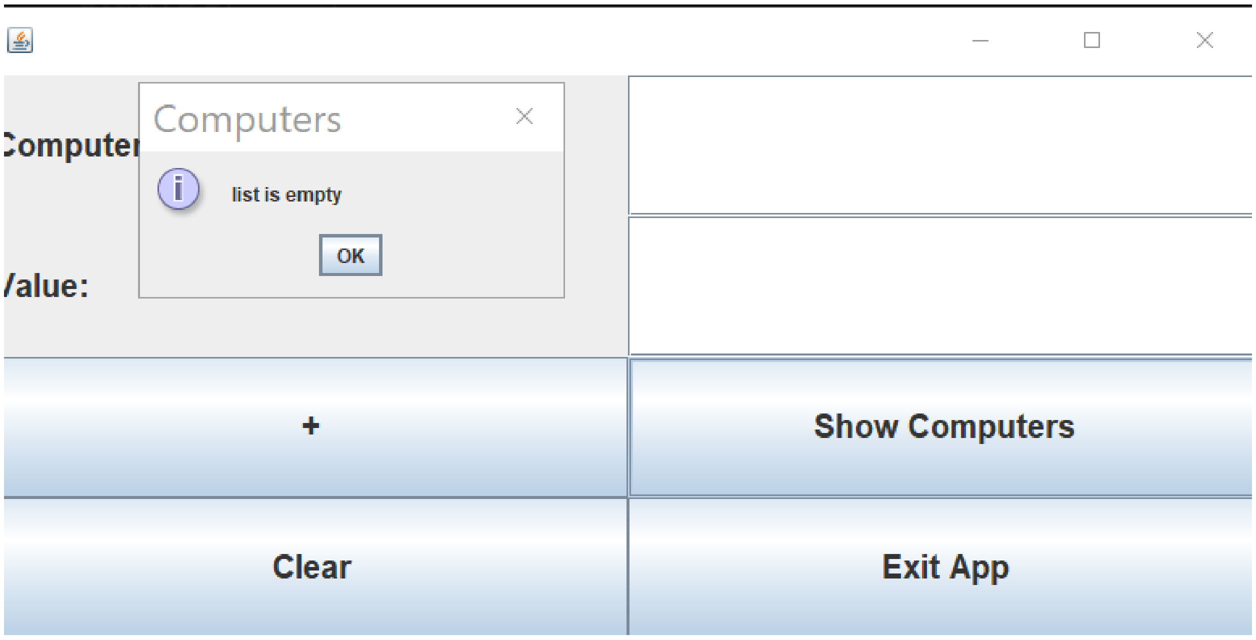
Student ID:	219404232
Name:	I Marembo
Course:	MT5675 - BSc Computer Science
Module:	CSI21M1 - Programming In Java

Assignment 2

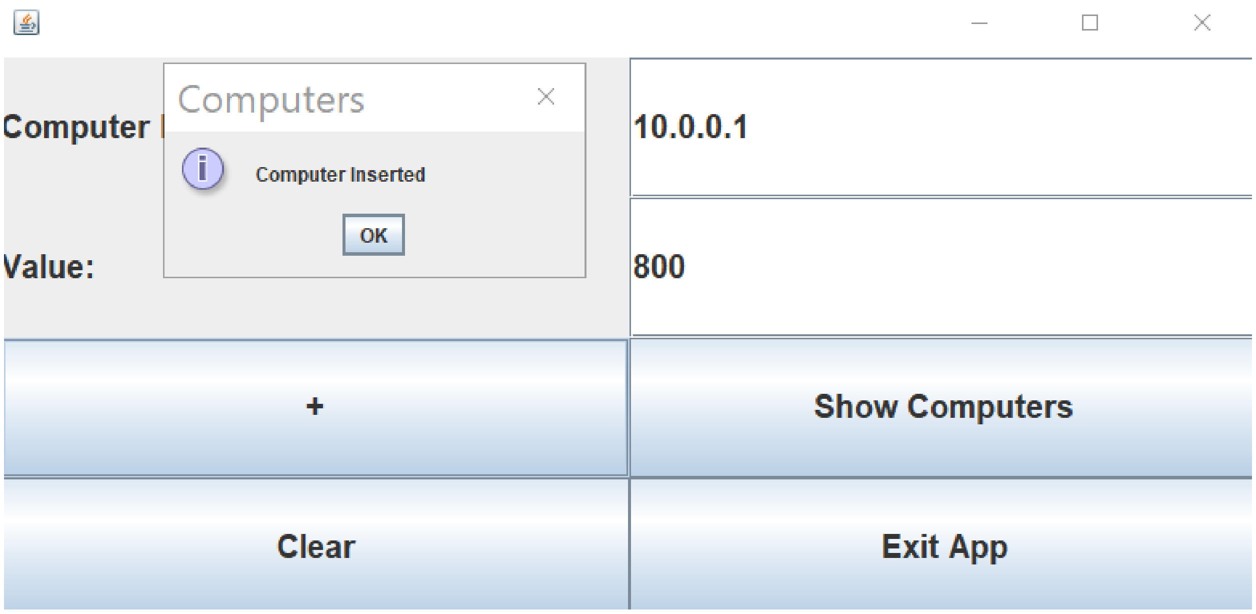
Source Code on GitHub
https://github.com/I-May-Public/219404232_CSI26M1_Assignment_2

Screenshots:

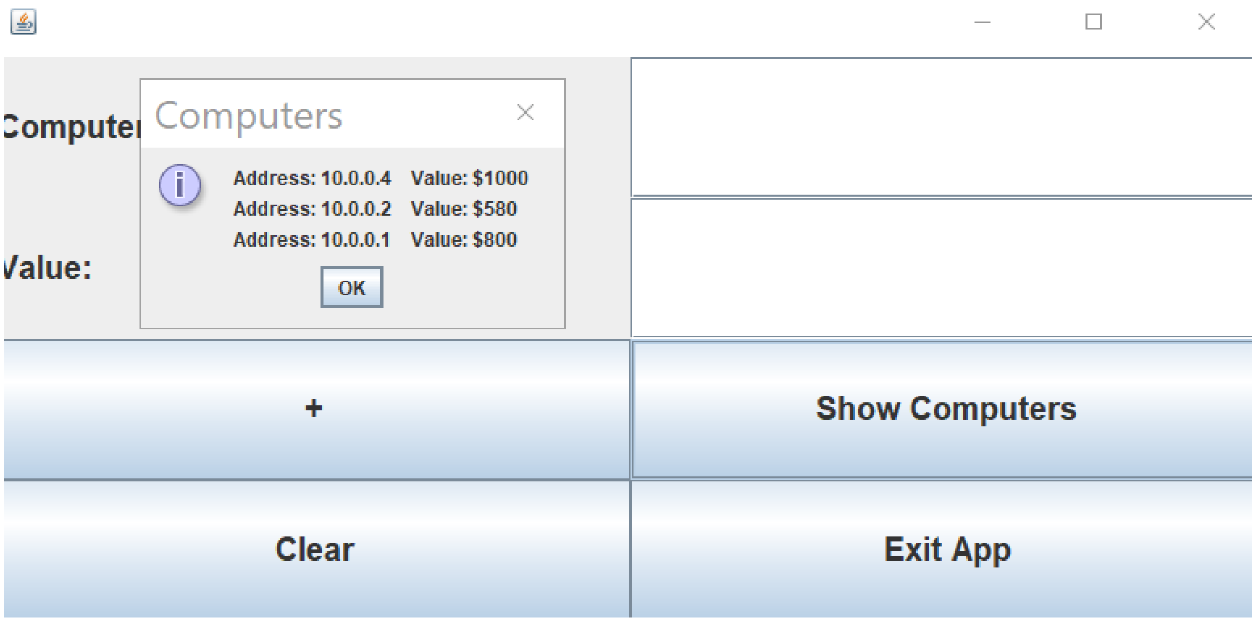




list is empty



Computer Inserted



```
Address: 10.0.0.4   Value: $1000
Address: 10.0.0.2   Value: $580
Address: 10.0.0.1   Value: $800
```

```java
public class Computer {

   String m_ip_address;
   String m_value;

   public Computer(String ip_address, String value) {
      SetIPAddress(ip_address);
      SetValue(value);
   }

   private void SetValue(String value) {
      this.m_value = value;
   }

   private void SetIPAddress(String ip_address) {
      this.m_ip_address = ip_address;
   }

   public String GetValue() {
      return this.m_value;
   }

   public String GetIPAddress() {
      return this.m_ip_address;
   }

   // Return Computer as a String
   @Override
   public String toString() {
      return "Address: " + GetIPAddress() + "    "
            + "Value: $" + GetValue() + "\n";
   }
}
```

```java
import javax.swing.JFrame;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JOptionPane;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.util.ArrayList;

public class ComputerSwingApp {
   static ArrayList<Computer> computers = new ArrayList<Computer>();

   // Parse List For Message Output
   static String ParseList(ArrayList<Computer> list, int offset) {
      int last = list.size() - 1;
      if (!list.isEmpty()) {
         if ((last - offset) > 0) {
            return list.get(last - offset).toString() + ParseList(list,
offset + 1);
         } else
            return list.get(last - offset).toString();
      } else
         return "list is empty";
   }

   // Append Computer to ArrayList
   static void AddComputer(Computer computer) {
      computers.add(computer);
      JOptionPane.showMessageDialog(null, "Computer Inserted", "Computers",
JOptionPane.INFORMATION_MESSAGE);
   }

   // Clear all Computers from ArrayList
   static void ClearComputers() {
      computers.clear();
   }

   // List All Computers in Dialog Box
   static void ShowComputers() {
      JOptionPane.showMessageDialog(null, ParseList(computers, 0),
"Computers", JOptionPane.INFORMATION_MESSAGE);
   }

   // Run App Window
   static void initialize() {
      // Labels
      JLabel address_label = new JLabel("Computer Lab IP Address:");
      JLabel value_label = new JLabel("Value:");

      // Input Fields
      JTextField address_input = new JTextField("");
      JTextField value_input = new JTextField("");

      // Buttons
      JButton add_btn = new JButton("+");
      JButton clear_btn = new JButton("Clear");
      JButton show_btn = new JButton("Show Computers");
```

```
        JButton exit_btn = new JButton("Exit App");

        // Set Fonts
        Font font = new Font("Arial", Font.BOLD, 20);
        address_label.setFont(font);
        value_label.setFont(font);

        address_input.setFont(font);
        value_input.setFont(font);

        add_btn.setFont(font);
        clear_btn.setFont(font);
        show_btn.setFont(font);
        exit_btn.setFont(font);

        // Frame
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(800, 400);
        frame.setVisible(true);
        frame.setLayout(new GridLayout(4, 2 /* , 10, 10 */));

        // Compose Frame
        frame.add(address_label);
        frame.add(address_input);

        frame.add(value_label);
        frame.add(value_input);

        frame.add(add_btn);
        frame.add(show_btn);
        frame.add(clear_btn);
        frame.add(exit_btn);

        // Button Action Listeners
        // ADD BUTTON
        add_btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {

                AddComputer(new Computer(address_input.getText(),
value_input.getText()));
                address_input.setText("");
                value_input.setText("");
            }
        });

        // SHOW BUTTON
        show_btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                ShowComputers();

            }
        });
        // CLEAR BUTTON
        clear_btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                ClearComputers();
            }
        });
        // EXIT BUTTON
        exit_btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {

                System.exit(0);
            }
        });
    }

    public static void main(String[] args) {
        initialize();
    }

}

```