CIS 120 Final Exam — Appendices

A Java to OCaml Code

A.1 `Animal` interface and `Cat` class

```
public interface Animal {
    public String getName();
    public String distinguishingFeature();
}


public class Cat implements Animal {

    private String name;

    public Cat(String name) {
        this.name = name;
    }

    @Override
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String distinguishingFeature() {
        return "chase the mouse!";
    }
}
```

A.2 Higher-order list processing functions in OCaml

```
let rec transform (f: 'a -> 'b) (l: 'a list): 'b list =
 begin match l with
   | [] -> []
   | h :: t -> (f h) :: (transform f t)
 end

let rec fold (combine: 'a -> 'b -> 'b) (base: 'b) (l: 'a list) : 'b =
 begin match l with
   | [] -> base
   | h :: t -> combine h (fold combine base t)
 end
```

B Java Swing Code

```
1  class ColorBulb extends JComponent {
2      private int red, green, blue;
3
4      public ColorBulb() {
5          red = 0;
6          green = 0;
7          blue = 0;
8      }
9
10     @Override
11     public void paintComponent(Graphics gc) {
12         gc.setColor(new Color(red, green, blue));
13         gc.fillRect(0, 0, 300, 300);
14     }
15
16     @Override
17     public Dimension getPreferredSize() {
18         return new Dimension(300, 300);
19     }
20
21     public void setRed(int red) {
22         this.red = red;
23     }
24
25     public void setGreen(int green) {
26         this.green = green;
27     }
28
29     public void setBlue(int blue) {
30         this.blue = blue;
31     }
32 }
33
34 public class ColorChanger implements Runnable {
35
36     @Override
37     public void run() {
38         JFrame frame = new JFrame("Color Changer");
39         frame.setLayout(new BorderLayout());
40
41         ColorBulb color = new ColorBulb();
42         frame.add(color, BorderLayout.CENTER);
43
44         JPanel panel = new JPanel();
45         frame.add(panel, BorderLayout.SOUTH);
46
47         JLabel label = new JLabel("Red Color");
48         panel.add(label);
49
50         JTextField redTextField = new JTextField(5);
51         panel.add(redTextField);
52
53         JButton button = new JButton("Change Red!");
54         panel.add(button);
```

```
55
56          button.addActionListener(new ActionListener() {
57              @Override
58              public void actionPerformed(ActionEvent e) {
59                  String r = redTextField.getText();
60                  int red = Integer.parseInt(r);
61                  color.setRed(red);
62                  color.repaint();
63              }
64          });
65
66          color.addMouseMotionListener(new MouseAdapter() {
67              @Override
68              public void mouseMoved(MouseEvent e) {
69                  color.setGreen(e.getX() % 255);
70                  color.setBlue(e.getY() % 255);
71                  color.repaint();
72              }
73          });
74          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
75          frame.pack();
76          frame.setVisible(true);
77      }
78
79      public static void main(String[] args) {
80          SwingUtilities.invokeLater(new ColorChanger());
81      }
82  }
```

C Java Code For PathSet

C.1 SimpleCollection

```
public interface SimpleCollection<E> extends Iterable<E> {
  boolean isEmpty();
  void add(E x);
  boolean contains(Object o);
}
```

C.2 PathSet.java

```
import java.util.Iterator;
import java.util.List;

public class PathSet implements SimpleCollection<List<String>> {

    private PathSetNode root;

    public PathSet() {
        this.root = new PathSetNode();
    }

    @Override
    public boolean isEmpty() {
        return root.isEmpty();
    }

    @Override
    public void add(List<String> path) {
        root.add(path);
    }

    @Override
    public boolean contains(Object o) {
        @SuppressWarnings("unchecked")
        List<String> path = (List<String>) o;
        return root.contains(path);
    }

    @Override
    public Iterator<List<String>> iterator() {
        List<List<String>> lls = root.toListOfPaths();
        return lls.iterator();
    }
}
```

C.3 `PathSetNode.java` (excerpt)

```
public class PathSetNode {
    private TreeMap<String, PathSetNode> children;
    private boolean isLast;

    public PathSetNode() {
        this.children = new TreeMap<String, PathSetNode>();
        this.isLast = false;
    }

    public boolean isEmpty() {
        return (!this.isLast && children.isEmpty());
    }

    public void add(List<String> path) {
      /* TODO */
    }

    public boolean contains(List<String> path) {
      /* TODO */
      return false;
    }

    public List<List<String>> toListOfPaths() {
      /* TODO */
      return null;
    }
}
```