**Adapt Interface – refactoring**

public class CardComponent extends Container ...

public CardComponent(Card card,Explanations explanations) {

...

addMouseMotionListener(createMouseMotionAdapter());

}

private MouseMotionAdapter createMouseMotionAdapter() {

return new MouseMotionAdapter() {

public void mouseDragged(MouseEvent e) {

e.consume();

dragPos.x = e.getX();

dragPos.y = e.getY();

setLocation(getLocation().x+e.getX()-currPos.x,

getLocation().y+e.getY()-currPos.y);

repaint();

}

};

}

The Adapter is created by writing a creation method for the Adapter, in this case it is MouseMotionAdapter. The empty method mouseMoved is then deleted. The code from mouseDragged method can then be moved to the Adapter, and the declaration of the implementation can be removed. The CardComponent constructor must then be changed to call the new creation method.

**Testing**

Because unit tests cannot be created for mouse events, manual testing is required. The tester should expand the card by dragging their mouse over it. If the card successfully expands, the code can be considered successfully refactored.