Capsule group:

Ryne Bell, Michael Gerhardt, Cassandra Cox

ANIMATION WITH JAVA

What is Animation?

- Animation is quite simply redrawing an object as its location changes/
- It can be used for more interesting GUIs, as well as for animating algorithms.
  - Animating algorithms can often make them easier to observe, understand and debug.

Multithreading and Animation

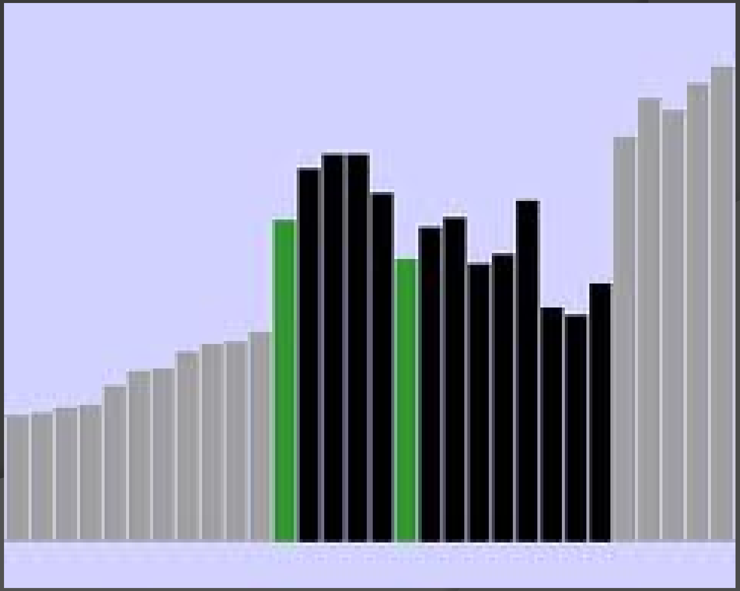
- Multithreading is used to animate multiple objects simultaneously.

// an easy way to create a new animation thread is to use this outline as a method in your class

```
public void startAnimation() {
        class AnimationRunnable implements Runnable {
                public void run() {
                        try {
                        // do whatever animation you’re doing with this thread
                        }
                        catch (InterruptedException exception) {
                        // catch the exception (the thread being
                        // interrupted) if you need to do so
                        }
                        }
                }
        Runnable r = new AnimationRunnable();
        Thread t = new Thread(r);
        t.start();
        }
```

Animating Algorithms

- In order to animate an algorithm, you must first decide what information you want to display.
- A basic algorithm animation is of a Selection Sort in action.
- The animation could, for example, show bars of the length of each element, and step slowly through the sort, Showing them move.



Animating Algorithms cont.

- In order to show the user/person debugging the algorithm what is going on, the algorithm must step slowly through its steps.

```
// an easy way to do this:
// steps is used to cause the delay to be proportional to the number of steps involved

Public void pause(int steps) throws Interrupted exception{
                component.repaint();
                Thread.sleep(steps*Delay);
}
```

Animating Algorithms cont.

- The StartAnimation method in an animated algorithm must both draw the visual representation of the algoritm, and step through it.

```
public void run(){
try{
// step through the algorithm however many steps, then animate it
}
catch (InterruptedException exception) {
                                                  }
pause(2); // pause between steps in the algorithm to allow the user to observe it
}
```