Filtering Data

Summary

Duke
UNIVERSITY

Interface Summary

```
public interface Filter {

    public boolean satisfies(QuakeEntry qe);

}
```

- Defining Interfaces :
  - Specify methods classes must have

Interface Summary

```
public class MinMagFilter implements Filter {
    private double magMin;
    public MinMagFilter(double min) {
        magMin = min;
    }
    public boolean satisfies(QuakeEntry qe) {
        return qe.getMagnitude() >= magMin;
    }
}
```

- Implementing Interfaces:
  - Write class with "`implements InterfaceName`"
  - Define all required methods

Interface Summary

```
public ArrayList<QuakeEntry>
filter(ArrayList<QuakeEntry> quakeData,
       Filter f) {
  ArrayList<QuakeEntry> answer
        = new ArrayList<QuakeEntry>();
  for(QuakeEntry qe : quakeData) {
    if (f.satisfies(qe)) {
      answer.add(qe);
    }
  }
  return answer;
}
```

- Using Interface Types:
  - Can use interface name as type
  - Can call methods in the interface

Interface Summary

```
Filter f = new MinMagFilter(4.0);
ArrayList<QuakeEntry> largeQuakes
        = filter(list, f);
```

- Type compatibility: use class as interface
  - Will still use methods in class: dynamic dispatch