Road Network Example

```
import java.util.LinkedList;

class Road {
    ...
}

class Intersection {
    ...
}

public class RoadNetwork {
    ...
}
```

OVERALL STRUCTURE

Road Network Example

```
/** Roads are one-way streets linking intersections
 *  @see Intersection
 */
class Road {
    float travelTime;         //measured in seconds
    Intersection destination; //where the road goes
    Intersection source;      //where the comes from
    // textual name of road is source-destination
}
```

Road Network Example

```
/** Intersections join roads
 *  @see Road
 */
class Intersection {
    String name;
    LinkedList <Road> outgoing = new LinkedList <Road> ();
    LinkedList <Road> incoming = new LinkedList <Road> ();
    // BUG: deal with type of intersections?
}
```

Road Network Example

```
/** Main Program
 *  @see Road
 *  @see Intersection
 */
public class RoadNetwork {
    public static void main(String[] args) {
        // BUG:  Need something testable here!
    }
}
```

Road Network Example

- Input data format:

```
intersection a
intersection b
road a b 5
road b a 11
```

- How do we process text?

- Java provides a tool, class `scanner`

Road Network Example

```
import java.io.File;
import java.util.Scanner;

/** Main Program
 *  @see Road
 *  @see Intersection
 */
public class RoadNetwork {
    public static void main(String[] args) {
        // BUG:  Need code to see if there is a file name
        Scanner sc = new Scanner(new File(args[0]));
        // BUG:  What if the file doesn't exist?
    }
}
```

Road Network Example

- The shell command:

```
[HawkID@serv15 ~/project]$ java RoadNetwork IowaCity.txt
```

- Consider this code from the main method:

```
Scanner sc = new Scanner(new File(args[0]));
```

- In context, this is now equivalent to this:

```
Scanner sc = new Scanner(new File("IowaCity.txt"));
```

Road Network Example

- A problem, from the Oracle page for class Scanner:

```
public Scanner(File source)
      throws FileNotFoundException
```

- We need a try-catch block:

```
try {
    Scanner sc = new Scanner( new File( args[0] ) );
    // BUG:  Now we can process the file here
} catch (FileNotFoundException e) {
    // BUG:  Complain that the file doesn't exist
}
```

Road Network Example

```
public class RoadNetwork {
    public static void main(String[] args) {
        if (args.length < 1) {
            // BUG:  Complain about a missing argument
        } else try {
            Scanner sc = new Scanner( new File(args[0]) );
Bad idea –  // BUG:  Now we can process the file here
        } catch (FileNotFoundException e) {
            // BUG:  Complain that the file doesn't exist
        }
    }
}
```

Road Network Example

```
public class RoadNetwork {
    private static void readNetwork( Scanner in ) {
        // Bug:  Details go here!
    }
    public static void main(String[] args) {
        if (args.length < 1) {
            // Bug:  Complain about a missing argument
        } else try {
Better idea –   readNetwork( new Scanner(new File(args[0])) );
        } catch (FileNotFoundException e) {
            // Bug:  Complain that the file doesn't exist
        }
    }
}
```

Road Network Example

```
private static void readNetwork( Scanner sc ) {
    while (sc.hasNext()) {
        // until the input file is finished
        string command = sc.next()
        if (command == "intersection") {
            new Intersection( sc );
        } else if (command == "road") {
            new Road( sc );
        } else {
            // Bug: Complain about unknown command
        }
    }
}
```

Road Network Example

```
    private static void readNetwork( Scanner sc ) {
        while (sc.hasNext()) {
            // until the input file is finished
            string command = sc.next()
            if (command == "intersection") {
Assumption –        new Intersection( sc );
            } else if (command == "road") {
                new Road( sc );
            } else {
                // Bug: Complain about unknown command
            }
        }
    }
```

Road Network Example

```
private static void readNetwork( Scanner sc ) {
    while (sc.hasNext()) {
        // until the input file is finished
        string command = sc.next()
        if (command == "intersection") {
            new Intersection( sc );
```

Assumption –

- `Intersection` constructor scans intersection description
- constructor complains about multiple definitions
- class `Intersection` maintains the intersection collection
- class `Intersection` offers a search tool by name
- class `Intersection` offers an iterator over all intersections

Road Network Example

```
private static void readNetwork( Scanner sc ) {
    while (sc.hasNext()) {
        // until the input file is finished
        string command = sc.next()
        if (command == "intersection") {
            new Intersection( sc );
        } else if (command == "road") {
            new Road( sc );
        } else {
            // Bug: Complain about unknown command
        }
    }
}
```

Assumption –

Road Network Example

```
private static void readNetwork( Scanner sc ) {
    while (sc.hasNext()) {
        // until the input file is finished
        string command = sc.next()
        if (command == "intersection") {
            new Intersection( sc );
        } else if (command == "road") {
            new Road( sc );
```

Assumption –

- Road constructor scans road description
- class Road maintains the road collection
- class Road offers an iterator over all roads

Epidemic Example

- A text format for specification:

```
pop 2500;       // population
house 4,3;      // household size average 4 +/– 3
jobs 0.25;      // 1/4 of the population has jobs
study 0.5;      // 1/2 of the population are students
```

- Probability distribution functions?
  - uniform distribution?
  - normal distribution?

    May as well hard code this for each, for now

    Empirical data for each random variable sets distribution

Fooling with class Scanner

```
import java.util.Scanner;

public class ScanTest {

    static final Scanner in = new Scanner( System.in );

    public static void main(String[] args) {
        System.out.print( in.next() );    // get string
        System.out.print( in.nextInt() ); // get an int
        System.out.print( in.next(";") ); // get a ";"
    }
}
```

Try it! Experiment!