Where are we?

```
class Road {
    float travelTime;    // in seconds
    Intersection destination;
    Intersection source;

    public Road( Scanner sc ) {
        // BUG: What goes here?
    }
}
```

A Constructor to make new Roads

```
public Road( Scanner sc )
    string sourceName = sc.next();
    string dstName = sc.next();
    travelTime = sc.nextFloat();
    sc.nextLine();

    // BUG: Set source from sourceName
    // BUG: Set destination from dstName
    // BUG: Create list of Roads
}
```

Extreme Programming

- Monday – develop (augment) specifications
- Tuesday – develop (augment) tests
- Wednesday – code
- Thursday – test and debug
- Friday – assess, discard week’s work if failure

Develop entire project in one-week steps

Development – backtrack search for solution

The Waterfall Model

Complete specification – the customer

Design the project – the architect

Implement – the building contractor

Verify – a testing agency

Maintain – service organizations

Water does not flow uphill

Even subcontract each stage to different vendor

Hierarchicy of Virtual Machines

- Hardware (at the bottom) supports
- An operating system which supports
- A programming language which supports
- General purpose libraries that support
- Special purpose components that support
- Higher level components that support
- The application

Development – work from the bottom up

Transparency

Each layer in the hierarchy may be

- Opaque – Blocks direct access to lower layers
  - safer, more secure, less efficient, less flexible
- Transparent – Permits such access
  - unsafe, insecure, potential efficiency, flexibility

In Java, `private`, `public` control transparency

Road Network

An incremental development plan

- First, read a description of the network,
  with just 1 type of intersection
  then just print it out to prove that it was read
- Then, add intersection types
  still debug by printing the network
- Then worry about simulation

Epidemic

An incremental development plan

- First, read a description of the community, with just 1 type of place
  then just print out the generated community
- Then, add types of places, one at a time
  still debug by printing the community
- Then worry about simulation

This means

For either model, we need to add

- A `toString()` method for each model class

- A `printModel()` static method in the main class
  for each instance of each model class
  print it out.