

## Initial Design

Initial design discussed equally as a group

Noun/Classes(possible)

Property (Tax variable)

Calculation class

## Verb/Methods

Owner can View a list of their properties

Department can view list of properties under certain headings

Register property

Pay tax




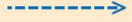
Get balance

Access previous records (csv file being read??)

Register owner

Find tax overdue and apply penalty

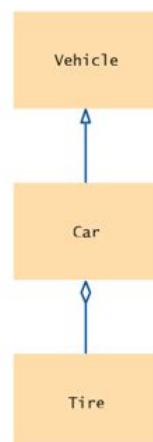
Be able to get statistics (only needs to be available to department) this is based off eircode

| Relationship             | Symbol  | Line Style | Arrow Tip |
|--------------------------|---|------------|-----------|
| Inheritance              |  | Solid      | Triangle  |
| Interface Implementation |  | Dotted     | Triangle  |
| Aggregation              |  | Solid      | Diamond   |
| Dependency               |  | Dotted     | Open      |

## Example

```
class Car extends Vehicle
{
    ...
    private Tire[] tires;
}
```

**Figure 6**  
UML Notation for  
Inheritance and Aggregation



Data fields for property class

Property owner(s)

Address

Postcode/Eircode

estimated market value

location category (City/Large town/Small town/Village/Countryside)

Principal private residence (yes/no)

Need exception handling for reading csv files

Probably need exception handling for user input = Built in handling in Scanner class

For eircode

Eircode is 7 characters long

First 3 is the routing key to identify the area

Possibly have two classes with main method

One of the classes has main method for running from command line and taking arguments from command line

The other class with main method implements the GUI