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

Methods:

toString()	For all the classes
ViewProperty(String property) (getter)	
RegisterProperty(String property) (setter)	
getPayment(double taxpaid)	
setPayment(double taxpaid)	
getBalance(double balance)	
setBalance(double balance)	
setRecord(String property, int owner)	
getRecord(String property, int owner)	
Overdue(double balance, double taxpaid)	

Arraylist:

- For Owner - that stores properties of that owner
- For the overdue payments for particular properties
- For Payment (payments made for property)

Responsibilities

Property class

- Read csv file
- Has a list of payments made on that property protected list
    - Protected list can be used within package
- Owner is just stored as a string
- Has a protected list of type payment that is available anywhere in the package
- Property owner(s) - owner can change hands or there could be new owners added??Co owners perhaps
- Potential change in owner ?
- setOwner method that allows to change the owner of the property then the property would have to be removed from the owners list of properties
- addOwner ? if there are now two owners of the property would then need to add this property to the owners list
- estimated market value can be updated - Property value decreases per year, *value depreciates*
- Principal private residence (yes/no) this status can change
- Has a list of unpaid payments **(overdues)** - (once paid, to be removed from overdue list)
- From overdue list (which of type payment and payment has a date field)can get year so that can calculate the time overdue
- Once the overdue payment has been made it is then removed from the overdue list
- Can add the payment made to the payment list where you would note the entire amount made in that payment this list of payment can be used later for balance statement if needed since the payments made would include the compounded
- Eircode is essentially a property id since each property has a unique eircode

Property History class

- Getting previous info on property
- Have a record of data that varies from year to year eg tax and the market value
- Will use list of payments and overdues from property

Department Personnel class

- Do statistics in here
- Statistics = total houses that paid in a particular area and the total tax located in a certain area
- Maybe list all registered in area and find a % of those paid
- Eircode routing thing
    - Eircode is 7 characters long
    - First 3 characters are the routing key to identify the area

Tax Calculator class

- Impose penalty for overdue tax will need access to history class for this
- Calculate tax for the given year

Payment class

- Records amount
- The year paid
- Fields would be the owner, property and amount

Owner class

- Pay tax and add to list of type payment in the property class - this list is protected
- View a list of their properties - print a list
- There is a list of type property within the owner class
- Needs to collab with property to know the tax for that property
- Owner can get balance statement for all their properties by just iterating through a list of type property and calling the balance statement method in the property class
- Owner has an id > Could have several people with the same name this id overcomes this problem

To Note

- In UML relationship diagram between property and payment is 1 or many
- Cause even if tax not actually paid the tax amount is still being recorded as a payment object
- Meaning a payment object is going to be used regardless

- We are assuming this system is only used by department personnel or owners already registered
    - We are assuming that no new owners are added by department - not in spec
- **Need exception handling for reading csv files and also for incorrect user input**
- Probably need exception handling for user input = Built in handling in Scanner class
- Append new property into the property csv files somewhere
- 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