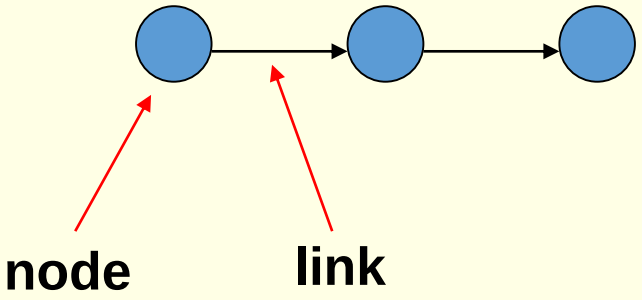
Anglia Ruskin University

THE LINKED LIST

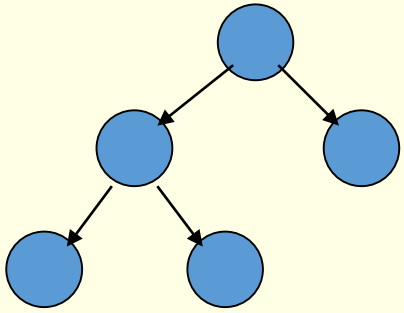
Many areas of Computer Science utilise

dynamic data structures

lists
(e.g stack, heap)



Binary trees



Anglia Ruskin University

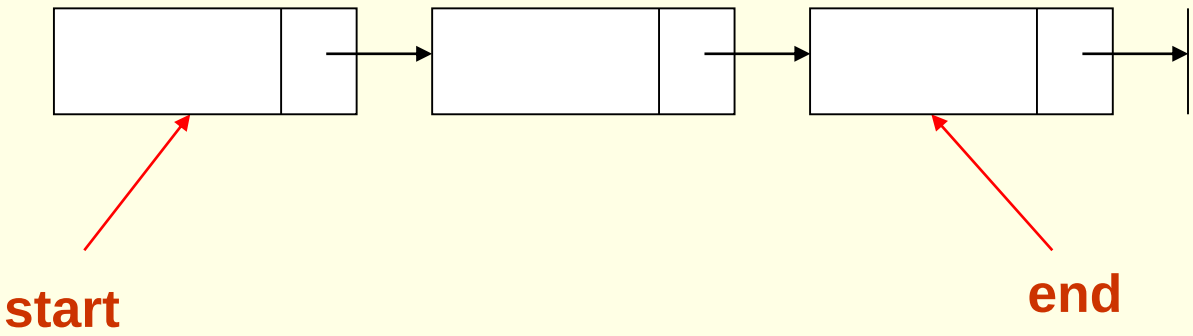
Eg. A linked list of 'Towns'

Define the Node class


```
class TownNode
{
    private string townName;
    private TownNode next;

    public TownNode( string name )
    {
        townName = name;
    }

    public TownNode getNext()
    {    return next;    }

    public string getName()
    {    return name;  }

    public void setNext( TownNode current )
    {    next = current;  }
}
```

Anglia Ruskin University

Define the List class



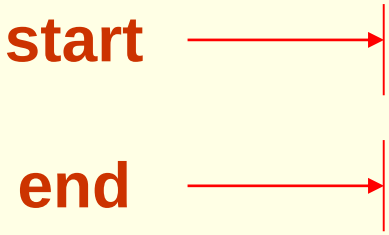

```
class TownList
{
    private TownNode start;
    private TownNode end;

    public TownList( ) {….}
    void addTownAtEnd( string name ) {….}
    void listAllNames( ) {….}
}
```

Anglia Ruskin University

1. Create a new ‘empty’ linked list

// create an empty list in Main()

```
TownList  towns = new TownList( );
```



// Implementation

```
public TownList( )
{
    start = null;
    end = null;
}
```

Anglia Ruskin University

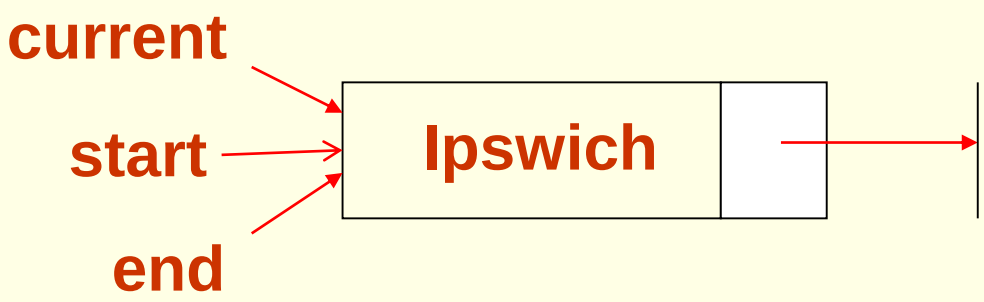
2. Add the first node to the list

// call method addTownAtEnd(string)

towns.addTownAtEnd(“Ipswich”);

// Implementation

```
public addTownAt End( string name )
{
    TownNode current  = new TownNode ( name );
     if ( end == null )
     {
         start = current;
         end = current;
     }
     else
     {….}
```



Anglia Ruskin University

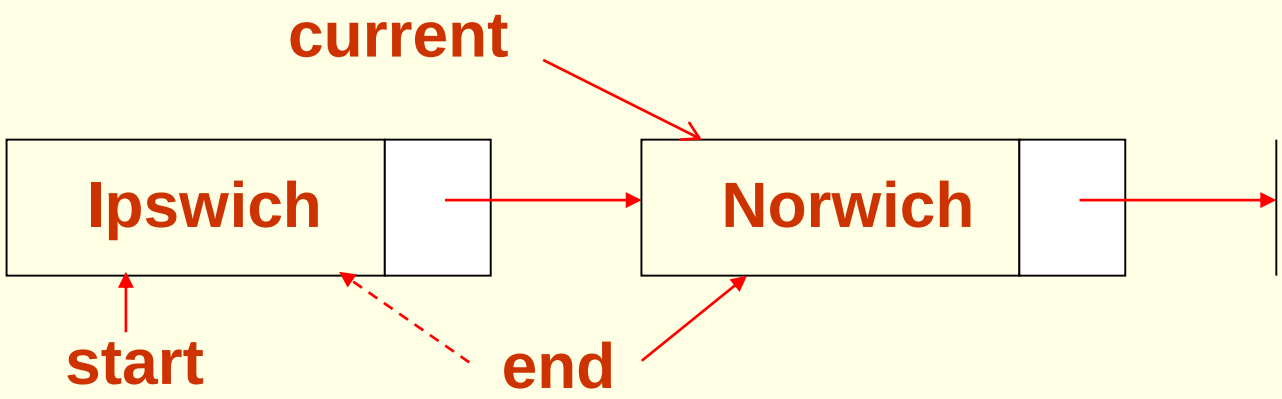
3. Add node to end of list

// call method addTownAtEnd()

towns.addTownAtEnd(“Norwich”);

// Implementation

```
if(….) {….}
else
{
    end.setNext( current );
    end = current;
}
```



Anglia Ruskin University

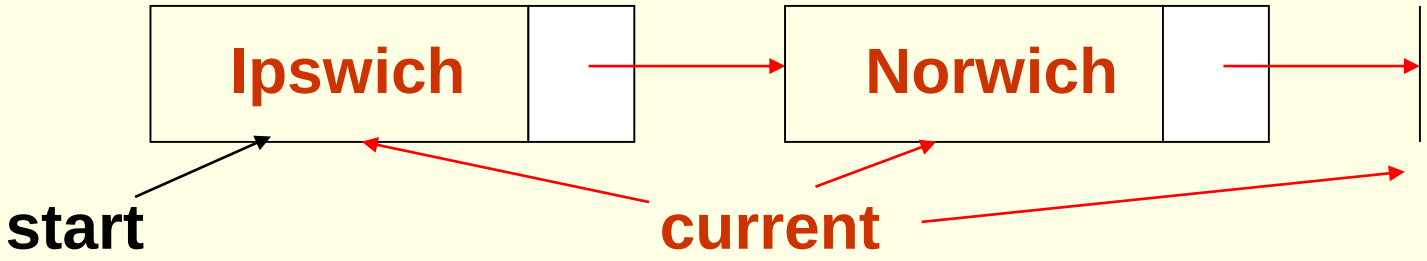
4. Traverse the list

// call method listAllNames()

```
towns.listAllNames( );
```

// Implementation

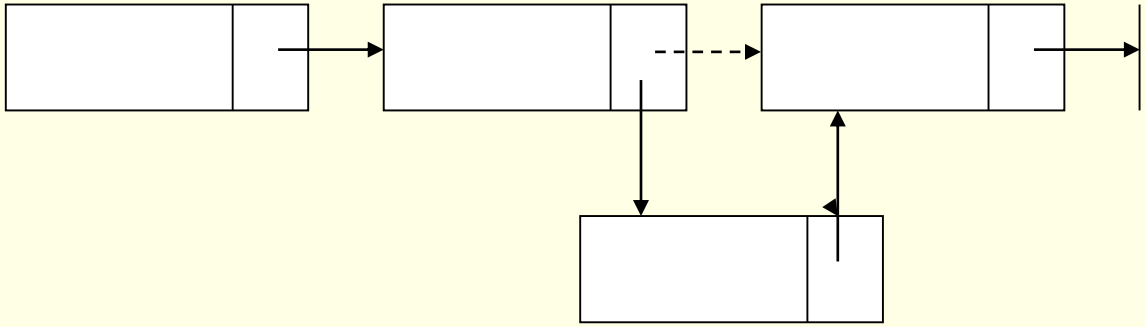
```
public void listAllNames( )
{
    TownNode current  = start;
    while( current  != null)
    {
       Console.WriteLine("Town is " + current.getName() );
       current = current.getNext();
    }
}
```



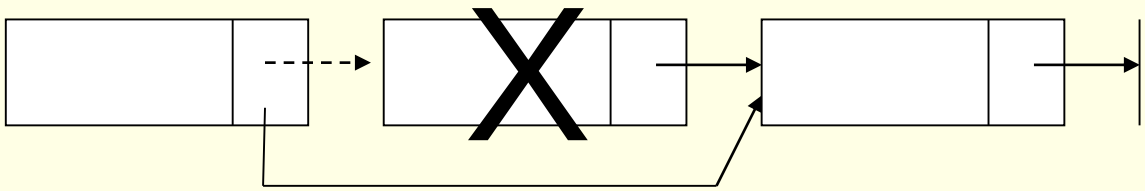
Anglia Ruskin University

More complicated operations

Insertions:



Deletions:



Anglia Ruskin University

The C# Standard Template Library

STL provides ‘off-the-shelf’ data structures including **containers**

- list - doubly linked list
- queue - FIFO
- stack - LIFO

Anglia Ruskin University

Assignment: one possible approach

1) Create all rooms
2) Set links
3) Start game (loop?)

Room
- n: Room - s: Room - e: Room - w: Room - …..?
+ Room() + setN(Room) + getN() : Room + ….?

Enhancing the assignment

- Implement inheritance
- Allow different maze configurations to be used (store in application, in separate file, and/or generate randomly)