Assignment 1: Software Design 4.1 2018/19 (30% of final mark)

Task: Design & Build a Java application based on Linked Lists, Swing (GUI Builder), Aggregation and Inheritance.

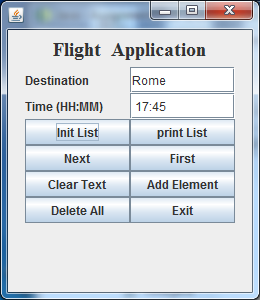
There is no limit to what you can do but it **must** involve some type of connection between Objects using Aggregation,and Inheritance e.g. Time & Journey and Flight as shown below and then a Swing GUI built using the GUI builder or directly with Swing which links to Linked List (collection) of these Objects.

In addition you could have :

* More Data, More Functionality than example given
* The GUI would include an Opening Frame, then click on a button to open the main frame shown
* Any further improvements you want to include

By next Fri 26/10/18 (10PM) must submit a brief specification of the application you are going to develop.

This might only contain the names of the main classed and a drawing (wireframe,word etc) of what the main window might look like: <http://mashable.com/2010/07/15/wireframing-tools/> or Visio in MS Office



**class** Time{

**private** **int** hour,minute;

**public** Time(**int** h,**int** m){

hour=h; minute=m;}

**public** String toString(){

**return** " "+hour+":"+minute;

}

**public** **void** reset(**int** h,**int** m){

hour=h; minute=m;}

}

**class** Journey{

**private** String destination;

**public** Journey(String dest){

destination=dest;}

**public** String readDestination(){**return** destination;}

}

**class** Flight **extends** Journey{

**private** Time departure\_time;

**public** Flight next;

**public** Flight(String dest,**int** h, **int** m){

**super**(dest);

departure\_time=**new** Time(h,m);}

**public** String readTime(){**return** departure\_time.toString();}

**public** **void** resetFlight(**int** h,**int** m){

departure\_time.reset(h,m);

}

**public** **void** print(){

System.*out*.print("["+**super**.readDestination()+ " : "

+ departure\_time+"]->");

}

}

See Very Basic Example in Moodle and it should be more complex than this.

Preliminary (brief) Spec to be stored in Moodle by 10PM Friday 26th October, 2016

Please Submit Writeup & Source Code to Moodle as a zipped file, it must include:

* Brief textual description of completed system
* Key classes and some code snippets of key parts of code
* Screen Shots
* Conclusions, Description of various problems encountered during development
* Source Code (.java files) must also be stored

Date: Final report and code stored on Moodle - Monday, 12th November, 2018

Marking Scheme:

1. Initial Specification, including how closely final system matches Spec (15%)
2. Final Specification, how well it’s presented (15%)
3. Completed Application (70%)

Marking Matrix to follow