

COMPSCI 230: Assignment 2

In this assignment, you will create a Swing application in Java. This application will read and modify CSV files. This will give you hands-on experience working with Java in *Eclipse* to develop a Swing application.

This assignment is due on **Monday 3 October, 2016, 11:59pm** and is worth 5% of your final course marks.

Estimated time to complete this assignment is 15-20 hours.

Background

The ambulance service for the city of Chromatropolis is in the process of updating their computer systems. One of their challenges is where the ambulances should be stationed in order to provide the best coverage. They are implementing a new system that will allow them to simulate ambulances picking up patients and delivering them to Chromatropolis Hospital.

Your task for this assignment is to build an application for entering the data for the simulation. The next assignment will then use the data from these files to run the simulation.

Submission

To prepare your solution for submission, ensure it is working on the lab computers. Provide an Eclipse project with code files that executes straight away. Your code should not depend on any external packages, e.g. it needs to compile and execute with the standard JAVA SDK and JRE.

Submit via the Assignment Dropbox (). Create a single .zip file of the entire project directory and submit that **one** file via the Dropbox. Note:

- Ensure you submit for the correct course.
- If you resubmit, please include **all** files in the .zip file of your re-submission.

Requirements

The application will provide a GUI interface, built in Java Swing, through which the user can add and remove ambulances and patients. Data for this application is storing in two CSV files called "ambulances.csv" and "patients.csv". Your application will need to load and parse these files, and save new data as needed.

Ambulance have the following attributes:

- ID: the identifier of the ambulance (a character followed by a one to three digit number).
- Location: the location of the ambulance
 - X: the x-coordinate (a number from 0 to 100)
 - Y: the y-coordinate (a number from 0 to 100)
- Status: the current status of the ambulance (one of 'At Station', 'Responding', 'At Scene', 'Transporting', 'At Destination', 'Returning')
- Patient: the identifier of the patient assigned to the ambulance (a number, can be empty)

The following is an example ambulances.csv file:

```
"id","x.location","y.location","status","patient"
"A7",17,25,"Responding",2
"A42",9,31,"At Scene",1
"A117",1,1,"At Station",
```

Patients have the following attributes:

- ID: the identifier of the patient (a number, auto-assigned by the system).
- Location: the location of the patient
 - X: the x-coordinate (a number from 0 to 100)
 - Y: the y-coordinate (a number from 0 to 100)
- Status: the current status of the patient (one of 'Pending', 'Assigned', 'Transporting', 'Completed')
- Ambulance: the ambulance assigned to the patient (a string, can be empty)

The following is an example patients.csv file:

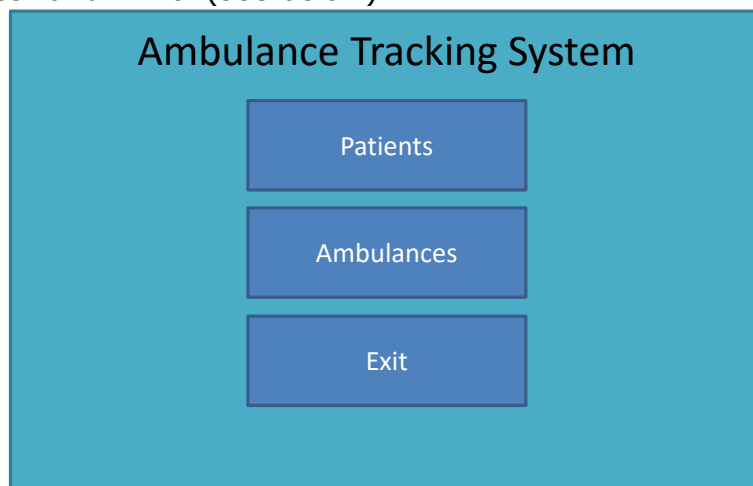
```
"id","x.location","y.location","status","ambulance"
1,9,31,"Transporting","A42"
2,27,40,"Assigned","A7"
3,82,12,"Pending",
```

A) Creating the assignment project

1. Create your project with *Eclipse*
2. Download the sample ambulances.csv and patients.csv files from Canvas
3. The main class is *AmbulanceTrackerApp*

B) Swing interface

1. Menu
 - a. This screen is the entry point for the application, it has three buttons: "Patients", "Ambulances" and "Exit" (see below)



- Clicking on the "Patients" button will display the **Patient List** screen
 - Clicking on the "Ambulances" button will display the **Ambulance List** screen
 - Clicking on the "Exit" button will end the application
2. Patient List
 - a. This screen displays all the patients, it has a list of all the patients and two buttons: "Back" and "Add New" (see below)

ID	Location	Status	Ambulance
1	(9, 3)	Transporting	A42
2	(27, 40)	Assigned	A7
3	(82,12)	Pending	-

Back Add New

- The list of patients displays the ID, location, status, and assigned ambulance of all the patients in the system
- Clicking on an patient will display the **Add/Edit Patient** screen with the clicked patients's details loaded
- Clicking on the "Back" button will display the **Menu** screen
- Clicking on the "Add New" button will display the **Add/Edit Patient** screen with no details in it

3. Add/Edit Patient

- This screen allows the user to edit the details of a patient, it has a label and input field for each field and two buttons: "Cancel" and "Save" (see below)

Patient: 1

ID: 1

Location: 9 3

Status: Transporting ▼

Ambulance: A7 ▼

Cancel Save

- The ID field is a read-only textbox containing the ID of the patient (for a new patient this will be automatically assigned by the system)
- The location fields allows the user to add/edit the location of the patient. The first textbox is the x- coordinate, the second the y-coordinate. Both inputs should only allow a number between 0 and 100.
- The status allows the user to set the status of the patient. It is a dropdown combobox with the following values: 'Pending', 'Assigned', 'Transporting', 'Completed'. The user cannot add new values to this list.
- The ambulance field allows the user to assign or unassign the patient to an ambulance. It is a dropdown combobox. The first (default) value is "None". The remaining values are populated from the Ambulance list in the system. These values must be in alphanumeric order. The user cannot add new values to this list.
- Clicking on the "Back" button will display the **Patient List** screen
- Clicking on the "Save" will validate and save the data:
 - If the data is invalid, a message will be displayed to the user with the details of the invalid data (e.g. "Location-x is a required field")

- If the data is valid, it will update their details in the file. For an existing patient, this will update the existing entry. For a new patient, this will add a new patient entry to the file.

4. Ambulance List

- This screen displays all the ambulances in the database, it has a list of all the ambulances and two buttons: "Back" and "Add New" (see below)

ID	Location	Status	Patient
A7	(17, 25)	Responding	2
A42	(9, 31)	At Scene	1
A117	(1, 1)	At Station	-

Back Add New

- The list of ambulances displays the ID, and station of all the ambulances in the database
- Clicking on an ambulance will display the **Add/Edit Ambulance** screen with the clicked ambulance's details loaded
- Clicking on the "Back" button will display the **Menu** screen
- Clicking on the "Add New" button will display the **Add/Edit Ambulance** screen with no details in it
- Bonus mark:* implement sorting when the user clicks on a column heading

5. Add/Edit Ambulance

- This screen allows the user to edit the details of an ambulance, it has a label and input field for each field and two buttons: "Cancel" and "Save" (see below)

Ambulance: A7

Ambulance ID:

Location:

Status: ▼

Patient: ▼

Cancel Save

- The ambulance ID field allows the user to add/edit the ID of the ambulance. This field is required; the first character must be an A followed by one or more digits (up to three).
- The location fields allows the user to add/edit the location of the ambulance. The first textbox is the x- coordinate, the second the y-coordinate. Both inputs should only allow a number between 0 and 100.
- The status allows the user to set the status of the ambulance. It is a dropdown combobox with the following values: 'At Station', 'Responding', 'At Scene', 'Transporting', 'At Destination', 'Returning'. The user cannot add new values to this list.

- e. The patient field allows the user to assign or unassign a patient to the ambulance. It is a dropdown combobox. The first (default) value is "None". The remaining values are populated from the Patient list in the system. These values must be in numeric order. The user cannot add new values to this list.
- f. Clicking on the "Back" button will display the **Ambulance List** screen
- g. Clicking on the "Save" will validate and save the data:
 - If the data is invalid, a message will be displayed to the user with the details of the invalid data (e.g. "Ambulance ID must consist of an A followed by a number")
 - If the data is valid, it will update their details in the file. For an existing ambulance, this will update the existing entry. For a new ambulance, this will add a new ambulance entry to the file.

C) Functionality

- When the system starts up, it will load all the data from the CSV files. This data should be parsed and held in in-memory data structures.
- When an ambulance is assigned or un-assigned to a patient, the associated patient entry should also be updated.
- When a patient is assigned or un-assigned to an ambulance, the associated ambulance entry should be updated.
- When the user clicks on Save from either the **Add/Edit Ambulance** or **Add/Edit Patient** screens, the updated data should be saved immediately to the files.

Mark Scheme

The marker will be looking for the following:

Item	Points
All screens correctly implemented, with data being populated from the database	10
Clicking on the button takes the user to the correct screen	5
Clicking on save correctly updates the CSV files (patients and ambulances)	5
Data is correctly validated prior to being saved (patients and ambulances)	5
Total (maximum mark)	25

Penalties

Marks will be deducted from your total for the following:

Penalty	Description
-5	Error in the application – applies if your program throws a run-time exception from any normal input by the marker
Up to -25	Hard coding – if the marker finds that you've inappropriately entered data or case-specific responses directly into your Java code rather than reading/writing to the CSV files
??	Late submission – the lecturer may accept late submissions by a scheme of penalties announced only after the deadline has passed. Note that extensions without penalty are possible for medical or compassionate grounds (not including workload in other courses, normal employer requirement or elective travel), or through request of University Counselling Services. The dropbox will remain open for a substantial period after the initial deadline to allow receipt of late submissions.