
Data Structures and Algorithms

COMP 103

2019-20

Semester 2

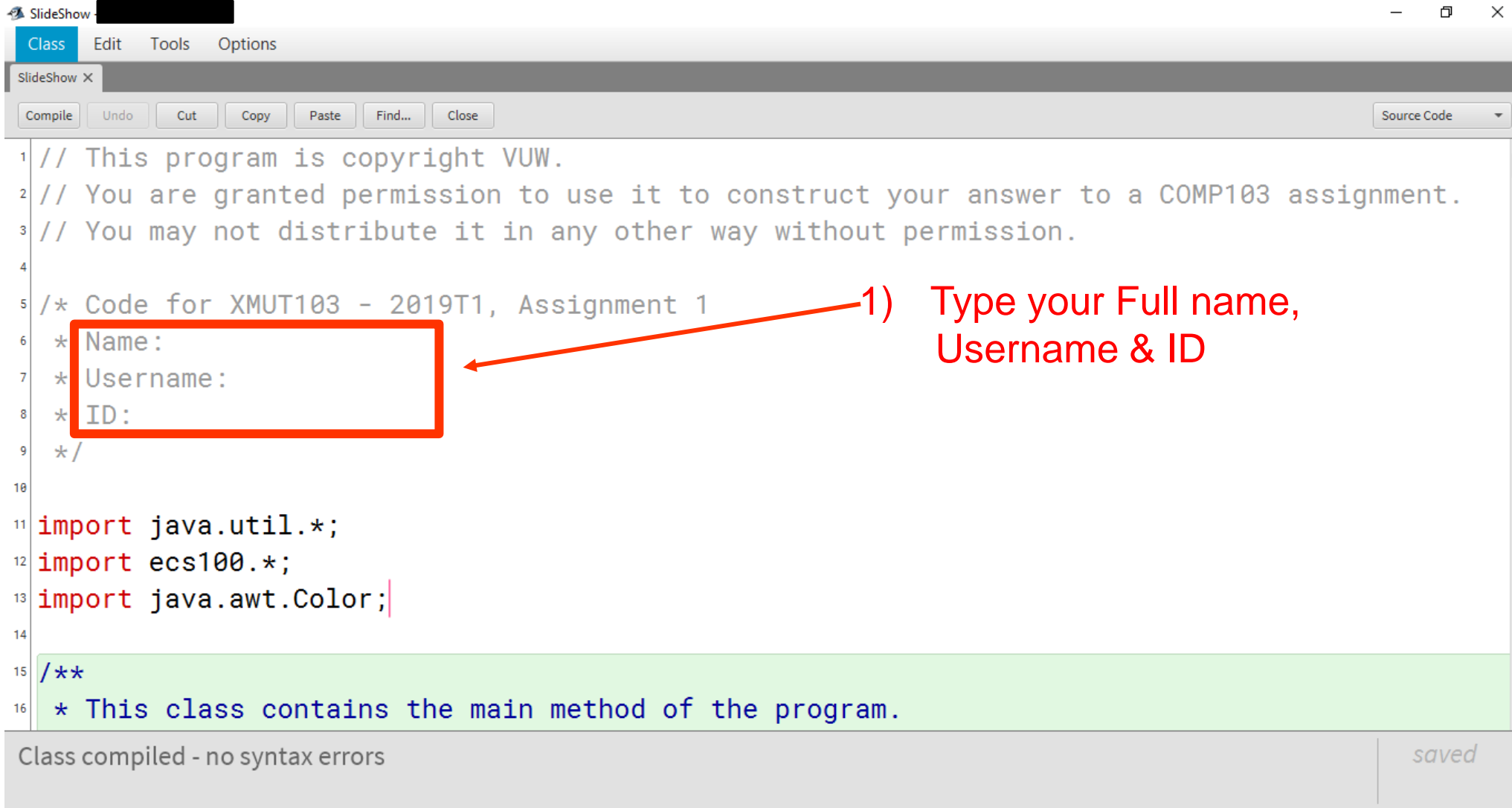
Lecture 07a

Dr. Kerese Manueli

kerese.manueli@ecs.vuw.ac.nz

Victoria University of Wellington

Assignment 1 - Feedback



The screenshot shows a Java IDE window titled "SlideShow" with a menu bar (Class, Edit, Tools, Options) and a toolbar (Compile, Undo, Cut, Copy, Paste, Find..., Close). The code editor contains the following text:

```
1 // This program is copyright VUW.
2 // You are granted permission to use it to construct your answer to a COMP103 assignment.
3 // You may not distribute it in any other way without permission.
4
5 /* Code for XMUT103 - 2019T1, Assignment 1
6  * Name:
7  * Username:
8  * ID:
9  */
10
11 import java.util.*;
12 import ecs100.*;
13 import java.awt.Color;
14
15 /**
16  * This class contains the main method of the program.
```

A red box highlights the comment block from line 6 to line 9. A red arrow points from the text "1) Type your Full name, Username & ID" to the red box.

Class compiled - no syntax errors saved

Assignment 1 - Feedback

```
1 // This program is copyright VUW.
2 // You are granted permission to use it to construct your answer to a COMP103 assignment.
3 // You may not distribute it in any other way without permission.
4
5 /* Code for XMUT103 - 2019T1, Assignment 1
6  * Name:
7  * Username:
8  * ID:
9  */
10
11 import java.util.*;
12 import ecs100.*;
13 import java.awt.Color;
14
15 /**
16  * This class contains the main m
17  */
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

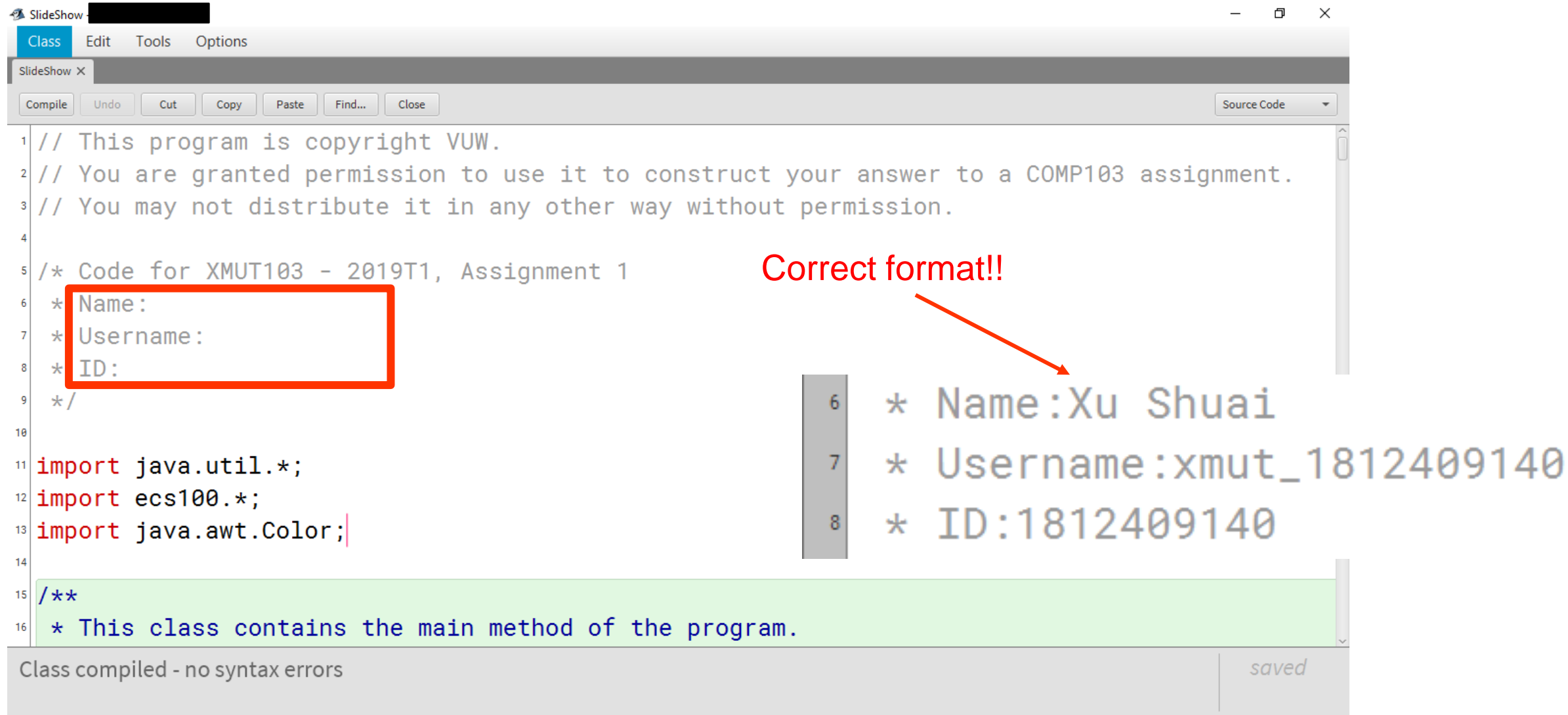
Class compiled - no syntax errors

saved

1) Type your Full name, Username & ID

2) Your Full name IS NOT Username. It looks like this XMUT_ID

Assignment 1 - Feedback



```
1 // This program is copyright VUW.
2 // You are granted permission to use it to construct your answer to a COMP103 assignment.
3 // You may not distribute it in any other way without permission.
4
5 /* Code for XMUT103 - 2019T1, Assignment 1
6  * Name:
7  * Username:
8  * ID:
9  */
10
11 import java.util.*;
12 import ecs100.*;
13 import java.awt.Color;
14
15 /**
16  * This class contains the main method of the program.
```

Correct format!!

```
6 * Name: Xu Shuai
7 * Username: xmut_1812409140
8 * ID: 1812409140
```

Class compiled - no syntax errors saved

[↑ XMUT103 home](#)[Course Outline](#)[Lecture Schedule](#)[Weekly Timetable](#)[Assignments](#)[Submission](#)[Your Marks](#)[People](#)[Java Resources](#)[Java documentation](#)[Tutor Space](#)[Assignment Admin](#)[Plagiarism Log](#)[School of Engineering and Computer Science](#) ▶ [Courses/XMUT103_2020T1](#) ▶ [Assignment2PartA](#)

Introduction to Data Structures and Algorithms

Assignment 2 Part A: Using Collections

- Due 6 May , 7pm

Resources and links

- Download [zip file](#) containing the necessary code and data.
- Java [Documentation](#)
- [Submit](#) your answers
- [Marks and feedback](#)

What To Hand In

- `WellingtonTrains.java`

Do not rename this file and do not rename any other files provided in the Template.

When you have submitted it, check that you can read the file listed on the submission page, and complete the submission process.

Wellington Trains (Weight: 2/3)

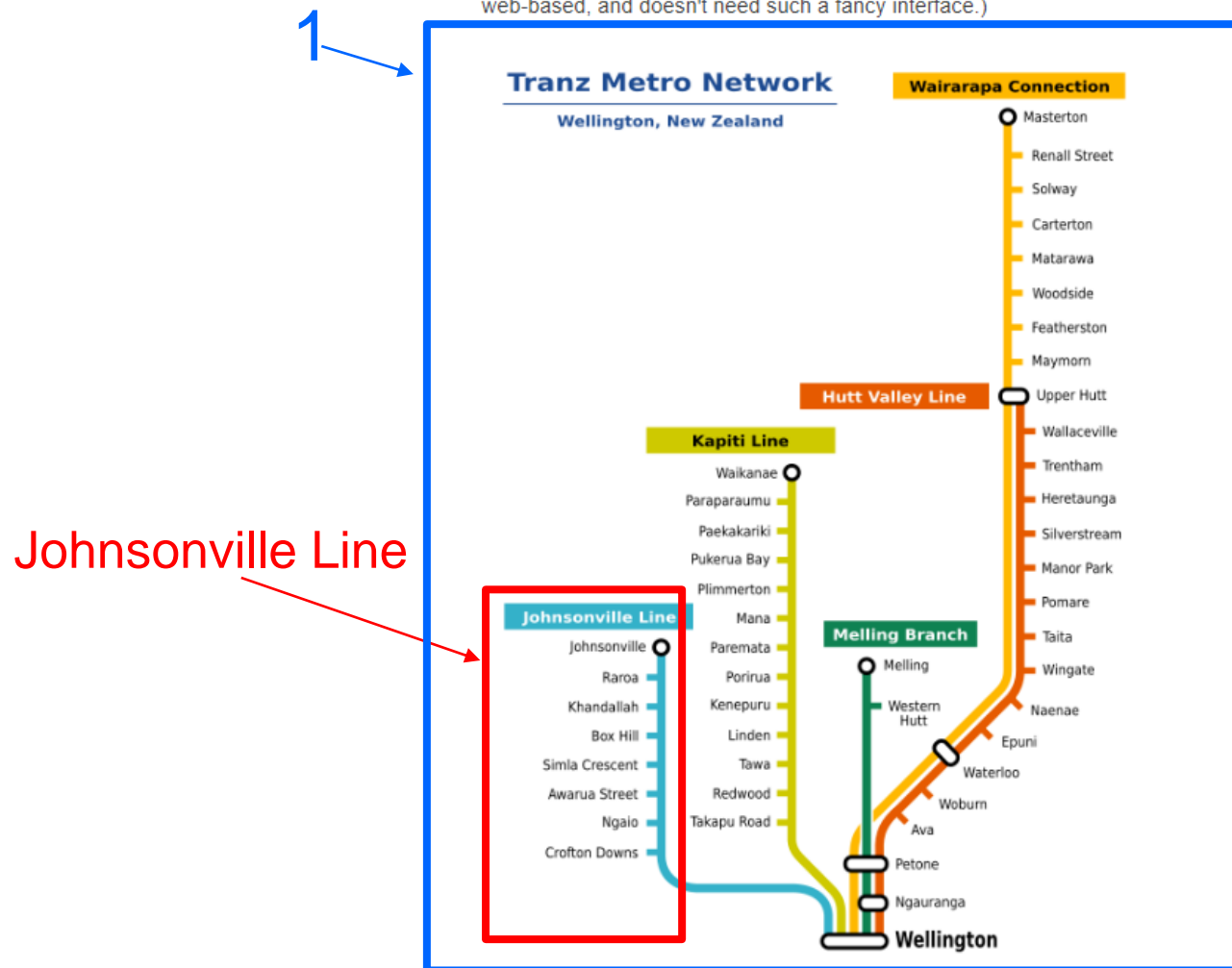
`WellingtonTrains` is a program to answer queries about Wellington train stations, train lines and the timetables for the train services on those lines.

(See <https://www.metlink.org.nz/#plan> for Metlink's equivalent program for the whole of the Wellington Regional Transport system. Your program is not

Wellington Trains (Weight: 2/3)

WellingtonTrains is a program to answer queries about Wellington train stations, train lines and the timetables for the train services on those lines.

(See <https://www.metlink.org.nz/#plan> for Metlink's equivalent program for the whole of the Wellington Regional Transport system. Your program is not web-based, and doesn't need such a fancy interface.)



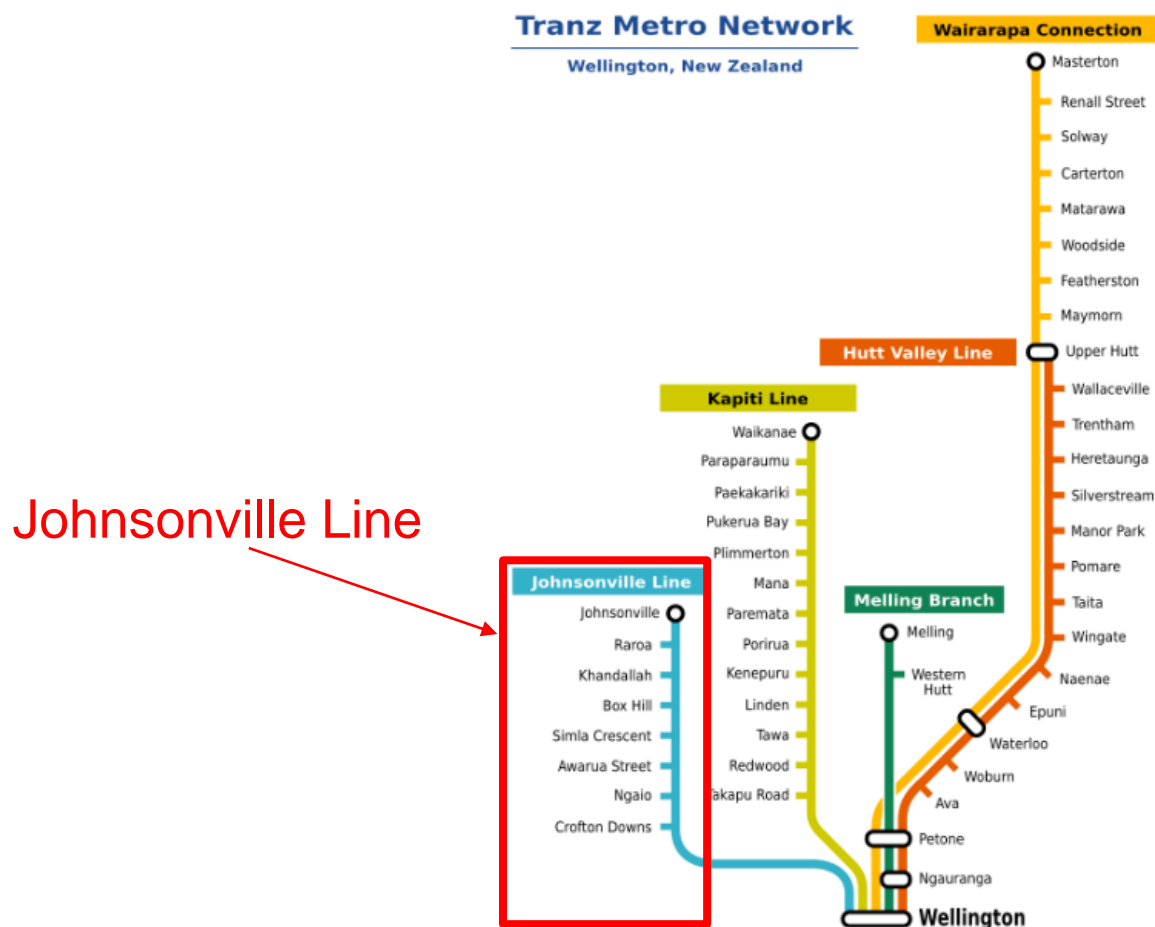
Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Wellington Trains (Weight: 2/3)

WellingtonTrains is a program to answer queries about Wellington train stations, train lines and the timetables for the train services on those lines.

(See <https://www.metlink.org.nz/#plan> for Metlink's equivalent program for the whole of the Wellington Regional Transport system. Your program is not web-based, and doesn't need such a fancy interface.)

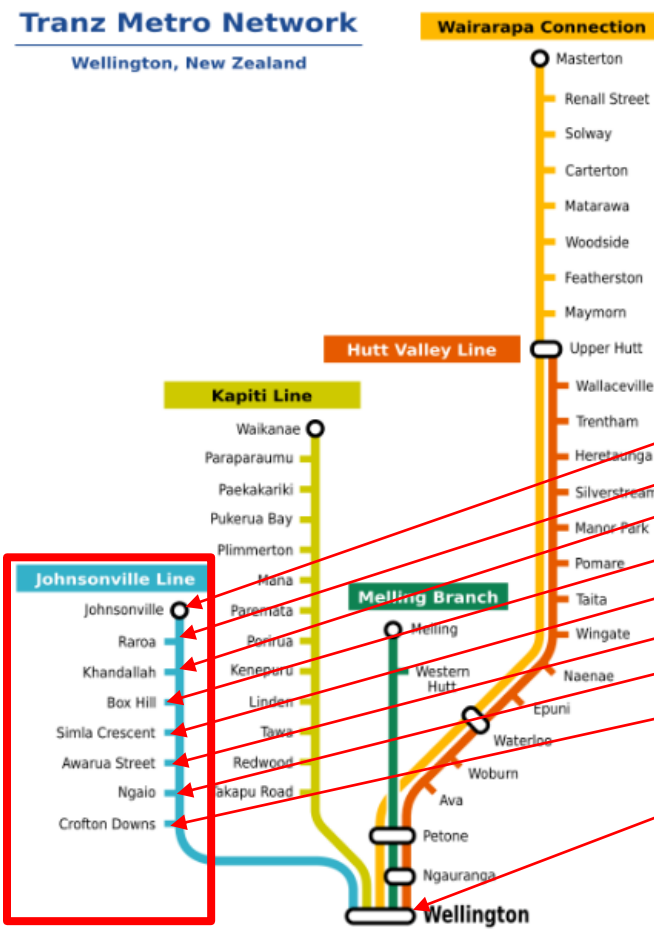


Name	Date modified	Type
fares.data	15/04/2020 3:51 PM	DATA File
geographic-map.png	15/04/2020 3:51 PM	PNG File
Johnsonville_Wellington-services.data	15/04/2020 3:51 PM	DATA File
Johnsonville_Wellington-stations.data	15/04/2020 3:51 PM	DATA File
Masterton_Wellington-services.data	15/04/2020 3:51 PM	DATA File
Masterton_Wellington-stations.data	15/04/2020 3:51 PM	DATA File
Melling_Wellington-services.data	15/04/2020 3:51 PM	DATA File
Melling_Wellington-stations.data	15/04/2020 3:51 PM	DATA File
stations.data	15/04/2020 3:51 PM	DATA File
system-map.png	15/04/2020 3:51 PM	PNG File
train-lines.data	15/04/2020 3:51 PM	DATA File
Upper-Hutt_Wellington-services.data	15/04/2020 3:51 PM	DATA File
Upper-Hutt_Wellington-stations.data	15/04/2020 3:51 PM	DATA File
Waikanae_Wellington-services.data	15/04/2020 3:51 PM	DATA File
Waikanae_Wellington-stations.data	15/04/2020 3:51 PM	DATA File
Wellington_Johnsonville-services.data	15/04/2020 3:51 PM	DATA File
Wellington_Johnsonville-stations.data	15/04/2020 3:51 PM	DATA File
Wellington_Masterton-services.data	15/04/2020 3:51 PM	DATA File
Wellington_Masterton-stations.data	15/04/2020 3:51 PM	DATA File
Wellington_Melling-services.data	15/04/2020 3:51 PM	DATA File
Wellington_Melling-stations.data	15/04/2020 3:51 PM	DATA File

Wellington Trains (Weight: 2/3)

WellingtonTrains is a program to answer queries about Wellington train stations, train lines and the timetables for the train services on those lines.

(See <https://www.metlink.org.nz/#plan> for Metlink's equivalent program for the whole of the Wellington Regional Transport system. Your program is not web-based, and doesn't need such a fancy interface.)



Johnsonville to Wellington Line

```

Johnsonville_Wellington-stations.data - Notepad
File Edit Format View Help
Johnsonville
Raroa
Khandallah
Box-Hill
Simla-Crescent
Awarua-Street
Ngaio
Crofton-Downs
Wellington
  
```


Stations, Lines, and Services:

Your program has to deal with three kinds of information:

- Train Stations
- Train Lines: a sequence of stations. For example, the Wellington_Johnsonville line, which starts at Wellington station, and ends at Johnsonville station, with 7 stations in between. To make things easier, we treat the inbound and outbound as two separate train lines, so the Wellington_Melling line is different from the Melling_Wellington line. (They have the same stations, but the sequence in one is the reverse of the sequence in the other.)
- Train Service: a schedule/timetable for a particular train running on a given train line. For example, the 11:32 train on the Wellington_Johnsonville line that leaves Wellington station at 11:32, leaves Crofton-Downs at 11:40, ... gets in to Johnsonville at 11:55.
A Train Service is specified by a sequence of times - the time that the train leaves the first station, followed by the times that the train gets to each of the remaining stations on the line. Note that some train services don't stop at every station on the line. If a train doesn't stop at a station, the corresponding time will be -1.

Queries

The basic queries the program should be able to handle are the following:

1. List all the stations in the region.
2. List all the train lines in the region
3. List the train lines that go through a given station
4. List the stations along a given train line
5. Print the name of a train line that goes from a station to a destination station. (The train line must go the correct direction.)
6. (Completion) Find the next train service for each line at a station after the specified time
7. (Completion) Find a trip between two stations (on the same line), after the specified time.
Find the train line, the time that next service on that line will leave the first station, the time that the service will arrive at the destination station, and the number of fare zones the trip goes through.
8. (Challenge:) More complex trips involve going from the first station to an exchange station on one train line, then going from the exchange station to the second station on another train line. (In larger cities, trips might require more than 2 train lines, but the Wellington region has a central hub, so you never need more than two segments for a trip.)
Find the best trip (earliest arrival) between a starting station and a destination station. If the trip requires two train lines, print out the starting time, the first train line, the exchange station, the arriving and departing times at the exchange station, the second train line, and the arrival time at the destination, and the total number of fare zones.

Data Files

There are a set of data files in the `data` sub directory that contain the information that your program needs to load and then use.

- `data/stations.data` contains one line of information about each station in the system: name, fare zone, and distance from the hub station. eg
`Crofton-Downs 3 4.9`

Note that there are no spaces in the station names - if the name has two parts, they are joined by a hyphen.

- `data/train-lines.data` contains the names of all the train lines, eg `Melling_Wellington`. Each line is named by the station at the beginning of the line and the station at the end of the line, joined by an underscore.
- For each train line, there are two files:
 - one with the sequence of stations on the train line (eg `data/Wellington_Johnsonville-stations.data`)
 - one with services (timetable) information (eg `data/Wellington_Johnsonville-services.data`)Each line of a `XXX_YYY-services.data` file has a sequence of times for one train service on the train line. The times are given in 24 hour time, with no ':' in the middle eg, 1437 is 2:37pm.

Data Structures

Your `WellingtonTrains` program will need to load the data from the files into data structures inside the program. A sensible way of organising the data inside your program will be to have

- a collection of `Stations` (a `Map`, indexed by the names of the stations)
- a collection of `TrainLines` (a `Map`, indexed by the names of the `TrainLines`)

For the Core of the assignment, you do not have to worry about the `Train Services` - just the `Stations` and `TrainLines`.

We have provided Java Classes for `Station`, `TrainLine`, and `TrainService` objects to store information about individual stations, train lines, and train services. Make sure that you read these classes carefully and work out how to use them. They each have constructors, and methods to add data to the object, and methods for accessing data in the object and a `toString()` method to return a string representation of the object. Note that a `TrainLine` contains inside it a collection of the `TrainServices` on that line, as well as a `List` of `Stations`, and a `Station` contains a collection of the `TrainLines` that go through that station.

Template Code.

In addition to the `Station.java`, `TrainLine.java`, and `TrainService.java` classes, we have provided a `WellingtonTrains.java` that contains just a few methods and fields. It contains a `main` method, a `doLoadData` method, and a `setupGUI` method for creating a user interface.

You will need to provide the methods invoked by the `setupGUI` method.

It is very important that you do not rename these methods

Suggested Algorithms for loading the data

To Create the Map of Stations:

- Read the data from `"data/stations.data"`
- For each line of the data,
 - construct a `Station` object using the data on the line
 - put the `Station` into the Map of Stations (the `Station` objects won't have any `TrainLine` information at this point)

To Create the Map of TrainLines:

- Read the list of train line names from `"data/train-lines.data"`
- For each train line name
 - Construct a `TrainLine` object
 - Put the `TrainLine` object into the Map of TrainLines
 - For each station name in `"data/XXX_YYY-stations.data"`
 - Look up the `Station` object in the Map of Stations,
 - Add the `Station` object to the stations in the `TrainLine` object,
 - Add the `TrainLine` object to the train lines in the `Station` object

To load the Train Service (timetable) information: [Completion only]

- Read the list of train line names from `"data/train-lines.data"`
- For each train line name
 - For each line (containing a list of times) in `"data/XXX_YYY-services.data"`

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

- Load the Station and Train Line data from the files into data structures. Methods to implement are `loadStationData` and `loadTrainLineData`.
- Be able to answer the following queries. Please follow these instructions carefully as we use automatic marking for marking your `WellingtonTrains` program
 - List all the stations in the region by writing a `listAllStations` method that **only** prints each station using `UI.println(station)`.
 - List all the train lines in the region by writing a `listAllTrainLines` method that **only** prints each train lines using `UI.println(line)`
 - List the train lines that go through a given station by writing a `listLinesOfStation` method that **only** prints each train line using `UI.println(line)`
 - List the stations along a given train line by writing a `listStationsOnLine` method that **only** prints each station using `UI.println(station)`
 - Print the name of a train line that goes from a station to a destination station. (The train line must go the correct direction.) Write a `checkConnected` method. The output should prints the name of the line followed by the number of fare zones, if there is a line connecting the two stations, or "No train line found" if the two stations are not connected. Note that this question is a bit tricky because it has to check that the first station comes before the destination station in the sequence of stations along the line. Please match the formt of the output.

checkConnected	
<p>Naenae and Woburn</p> <p><input type="button" value="All Stations"/> Upper-Hutt_Wellington 2 fare zones</p> <p><input type="button" value="All Lines"/></p> <p>Station <input type="text" value="Naenae"/></p> <p>Train Line <input type="text"/></p> <p>Destination <input type="text" value="Woburn"/></p> <p>Time (24hr) <input type="text"/></p>	<p>Naenae and Melling</p> <p><input type="button" value="All Stations"/> No train line found</p> <p><input type="button" value="All Lines"/></p> <p>Station <input type="text" value="Naenae"/></p> <p>Train Line <input type="text"/></p> <p>Destination <input type="text" value="Melling"/></p> <p>Time (24hr) <input type="text"/></p>

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

- Load the Station and Train Line data from the files into data structures. Methods to implement are loadStationData and loadTrainLineData.
- Be able to answer the following queries. Please follow these instructions carefully as we use automatic marking for marking your `WellingtonTrains` program
 - List all the stations in the region by writing a `listAllStations` method that **only** prints each station using `UI.println(station)`.
 - List all the train lines in the region by writing a `listAllTrainLines` method that **only** prints each train lines using `UI.println(line)`
 - List the train lines that go through a given station by writing a `listLinesOfStation` method that **only** prints each train line using `UI.println(line)`
 - List the stations along a given train line by writing a `listStationsOnLine` method that **only** prints each station using `UI.println(station)`
 - Print the name of a train line that goes from a station to a destination station. (The train line must go the correct direction.) Write a `checkConnected` method. The output should prints the name of the line followed by the number of fare zones, if there is a line connecting the two stations, or "No train line found" if the two stations are not connected. Note that this question is a bit tricky because it has to check that the first station comes before the destination station in the sequence of stations along the line.

Please match the format of the output.

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

- Load the Station and Train Line data from the files into data structures. Methods to implement are `loadStationData` and `loadTrainLineData`.

```
stations.data - Notepad
File Edit Format View Help
Wellington 1 0
Ngauranga 3 4.8
Crofton-Downs 3 4.9
Ngao 3 5.2
Awarua-Street 3 6
Simla-Crescent 3 6.9
Box-Hill 3 7.2
Khandallah 3 8
Raroa 3 9.2
Johnsonville 3 10.5
Petone 4 10.5
Takapu-Road 4 11.9
Western-Hutt 4 11.9
Ava 4 12.5
Redwood 4 13.1
Melling 4 13.5
Tawa 4 13.8
Woburn 4 14.4
Linden 4 14.9
Waterloo 4 15.5
Kenepuru 5 16.2
Epuni 5 16.5
Porirua 5 17.7
Naenae 5 18.3
Wingate 5 19.5
Taita 5 20.6
Paremata 6 21.9
Pomare 5 22
Mana 6 23.2
Manor-Park 6 23.7
Plimmerton 6 24.5
Silverstream 6 26.8
Heretaunga 6 28.2
Trentham 6 29.4
Pukerua-Bay 7 30.4
```

```
train-lines.data - Notepad
File Edit Format View Help
Johnsonville_Wellington
Wellington_Johnsonville
Melling_Wellington
Wellington_Melling
Waikanae_Wellington
Wellington_Waikanae
Masterton_Wellington
Wellington_Masterton
Upper-Hutt_Wellington
Wellington_Upper-Hutt
```


Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

- Load the Station and Train Line data from the files into data structures. Methods to implement are `loadStationData` and `loadTrainLineData`.

```
stations.data - Notepad
File Edit Format View Help
Wellington 1 0
Ngauranga 3 4.8
Crofton-Downs 3 4.9
Ngao 3 5.2
Awarua-Street 3 6
Simla-Crescent 3 6.9
Box-Hill 3 7.2
Khandallah 3 8
Raroa 3 9.2
Johnsonville 3 10.5
Petone 4 10.5
Takapu-Road 4 11.9
Western-Hutt 4 11.9
Ava 4 12.5
Redwood 4 13.1
Melling 4 13.5
Tawa 4 13.8
Woburn 4 14.4
Linden 4 14.9
Waterloo 4 15.5
Kenepuru 5 16.2
Epuni 5 16.5
Porirua 5 17.7
Naenae 5 18.3
Wingate 5 19.5
Taita 5 20.6
Paremata 6 21.9
Pomare 5 22
Mana 6 23.2
Manor-Park 6 23.7
Plimmerton 6 24.5
Silverstream 6 26.8
Heretaunga 6 28.2
Trentham 6 29.4
Pukerua-Bay 7 30.4
```

```
24 public class WellingtonTrains{
25     //Fields to store the collections of Stations and Lines
26     /*# YOUR CODE HERE */
27     private Map<String, Station> allStations = new HashMap<String, Station>();
```

You need 2 data structures.
ONLY 1 is shown above as an example!

```
train-lines.data - Notepad
File Edit Format View Help
Johnsonville_Wellington
Wellington_Johnsonville
Melling_Wellington
Wellington_Melling
Waikanae_Wellington
Wellington_Waikanae
Masterton_Wellington
Wellington_Masterton
Upper-Hutt_Wellington
Wellington_Upper-Hutt
```

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

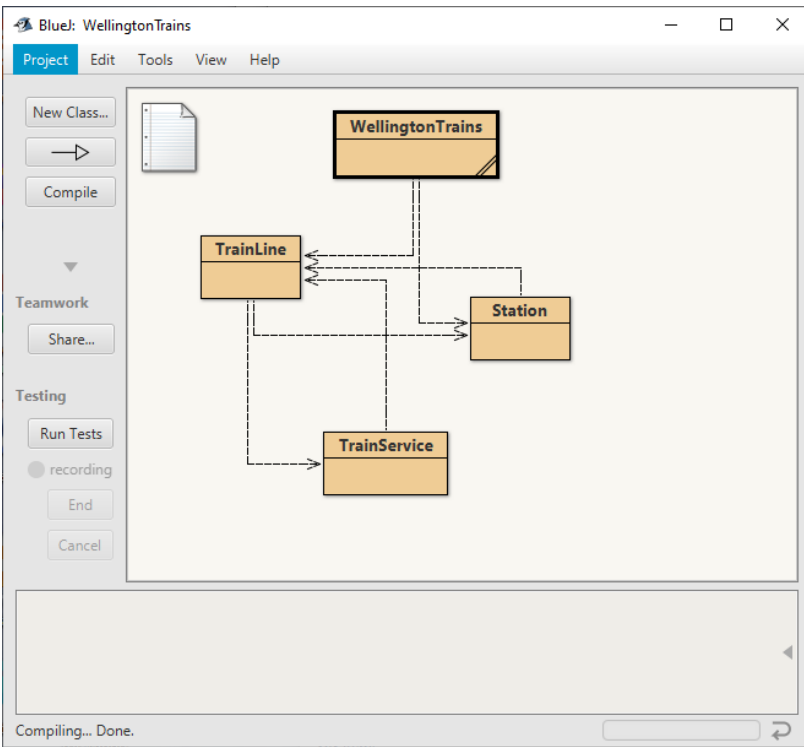
- Load the Station and Train Line data from the files into data structures. Methods to implement are loadStationData and loadTrainLineData.

```
stations.data - Notepad
File Edit Format View Help
Wellington 1 0
Ngauranga 3 4.8
Crofton-Downs 3 4.9
Ngaoi 3 5.2
Awarua-Street 3 6
Simla-Crescent 3 6.9
Box-Hill 3 7.2
Khandallah 3 8
Raroa 3 9.2
Johnsonville 3 10.5
Petone 4 10.5
Takapu-Road 4 11.9
Western-Hutt 4 11.9
Ava 4 12.5
Redwood 4 13.1
Melling 4 13.5
Tawa 4 13.8
Woburn 4 14.4
Linden 4 14.9
Waterloo 4 15.5
Kenepuru 5 16.2
Epuni 5 16.5
Porirua 5 17.7
Naenae 5 18.3
Wingate 5 19.5
Taita 5 20.6
Paremata 6 21.9
Pomare 5 22
Mana 6 23.2
Manor-Park 6 23.7
Plimmerton 6 24.5
Silverstream 6 26.8
Heretaunga 6 28.2
Trentham 6 29.4
Pukerua-Bay 7 30.4
```

```
train-lines.data - Notepad
File Edit Format View Help
Johnsonville_Wellington
Wellington_Johnsonville
Melling_Wellington
Wellington_Melling
Waikanae_Wellington
Wellington_Waikanae
Masterton_Wellington
Wellington_Masterton
Upper-Hutt_Wellington
Wellington_Upper-Hutt
```

```
WellingtonTrains - WellingtonTrains
Class Edit Tools Options
WellingtonTrains x
Compile Undo Cut Copy Paste Find... Close Source Code
41 }
42
43 /**
44  * Load data files
45  */
46 public void loadData(){
47     loadStationData();
48     UI.println("Loaded Stations");
49     loadTrainLineData();
50     UI.println("Loaded Train Lines");
```


Compiling & running the WellingtonTrains program



Click on the **All Stations** Button to produce the output as shown

* MENU *

- All Stations
- All Lines
- Station
- Train Line
- Destination
- Time (24hr)
- Lines of Station
- Stations on Line
- Stations connected?
- Next Services
- Find Trip
- Quit

All Stations in region:

Box-Hill (zone 3, 0 lines)
Naenae (zone 5, 0 lines)
Plimmerton (zone 6, 0 lines)
Maymorn (zone 8, 0 lines)
Mana (zone 6, 0 lines)
Epunui (zone 5, 0 lines)
Ngauranga (zone 3, 0 lines)
Woburn (zone 4, 0 lines)
Takapu-Road (zone 4, 0 lines)
Upper-Hutt (zone 7, 0 lines)
Johnsonville (zone 3, 0 lines)
Waterloo (zone 4, 0 lines)
Petone (zone 4, 0 lines)
Solway (zone 14, 0 lines)
Redwood (zone 4, 0 lines)
Featherston (zone 11, 0 lines)
Simla-Crescent (zone 3, 0 lines)
Melling (zone 4, 0 lines)
Linden (zone 4, 0 lines)
Masterton (zone 14, 0 lines)
Waikanae (zone 10, 0 lines)
Awarua-Street (zone 3, 0 lines)
Woodside (zone 12, 0 lines)
Wellington (zone 1, 0 lines)
Pukerua-Bay (zone 7, 0 lines)
Wallaceville (zone 7, 0 lines)
Western-Hutt (zone 4, 0 lines)
Paremata (zone 6, 0 lines)
Trentham (zone 6, 0 lines)
Raroa (zone 3, 0 lines)
Taita (zone 5, 0 lines)
Ava (zone 4, 0 lines)
Wingate (zone 5, 0 lines)
Matarawa (zone 13, 0 lines)
Crofton-Downs (zone 3, 0 lines)
Khandallah (zone 3, 0 lines)
Tawa (zone 4, 0 lines)
Silverstream (zone 6, 0 lines)
Heretaunga (zone 6, 0 lines)
Pomare (zone 5, 0 lines)
Ngaio (zone 3, 0 lines)
Paraparaumu (zone 9, 0 lines)
Manor-Park (zone 6, 0 lines)
Porirua (zone 5, 0 lines)
Paekakariki (zone 8, 0 lines)
Renall-Street (zone 14, 0 lines)
Carterton (zone 13, 0 lines)
Kenepuru (zone 5, 0 lines)

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

- Load the Station and Train Line data from the files into data structures. Methods to implement are `loadStationData` and `loadTrainLineData`.
- Be able to answer the following queries. Please follow these instructions carefully as we use automatic marking for marking your WellingtonTrains program
 - List all the stations in the region by writing a `listAllStations` method that **only** prints each station using `UI.println(station)`.
 - List all the train lines in the region by writing a `listAllTrainLines` method that **only** prints each train lines using `UI.println(line)`

```
60 public void setupGUI(){
61     UI.addButton("All Stations", this::listAllStations);
62     UI.addButton("All Lines", this::listAllTrainLines);
```

```
public void listAllTrainLines() {
}
```

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

Core:

For the Core, your program should

- Load the Station and Train Line data from the files into data structures. Methods to implement are `loadStationData` and `loadTrainLineData`.
- Be able to answer the following queries. Please follow these instructions carefully as we use automatic marking for marking your `WellingtonTrains` program
 - List all the stations in the region by writing a `listAllStations` method that **only** prints each station using `UI.println(station)`.
 - List all the train lines in the region by writing a `listAllTrainLines` method that **only** prints each train lines using `UI.println(line)`

Result of the All Lines button

The screenshot shows a Java Swing window titled '* MENU *'. It has three buttons: 'All Stations', 'All Lines', and 'All Services'. The 'All Lines' button is highlighted with a green arrow pointing from the text 'Result of the All Lines button'. Below the buttons are three text input fields labeled 'Station', 'Train Line', and 'Destination'. To the right of the buttons, a text area displays the output of the `listAllTrainLines` method, which is enclosed in a green box. The output is as follows:

```
All Train Lines in region:
-----
Johnsonville_Wellington (9 stations, 45 services)
Wellington_Johnsonville (9 stations, 45 services)
Wellington_Melling (5 stations, 23 services)
Wellington_Waikanae (14 stations, 58 services)
Melling_Wellington (5 stations, 23 services)
Wellington_Masterton (12 stations, 5 services)
Waikanae_Wellington (14 stations, 59 services)
Wellington_Upper-Hutt (17 stations, 57 services)
Masterton_Wellington (12 stations, 5 services)
Upper-Hutt_Wellington (17 stations, 54 services)
```

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

- List the train lines that go through a given station by writing a `listLinesOfStation` method that **only** prints each train line using `UI.println(line)`
- List the stations along a given train line by writing a `listStationsOnLine` method that **only** prints each station using `UI.println(station)`

* MENU *

All Stations

All Lines

Station

Johnsonville

Train Line

Destination

Time (24hr)

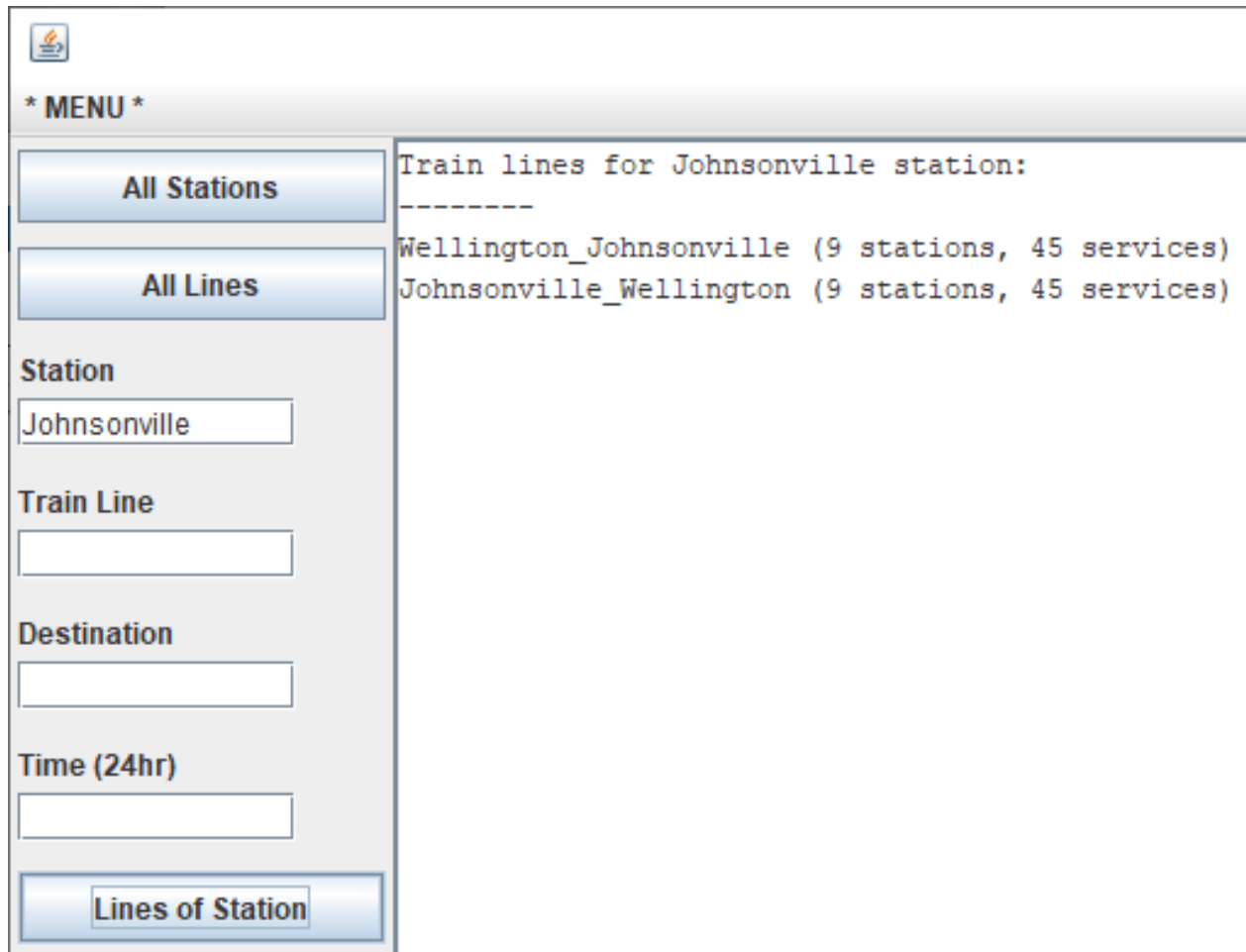
Lines of Station

Loaded Stations
Loaded Train Lines
Loaded Train Services

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

- List the train lines that go through a given station by writing a `listLinesOfStation` method that **only** prints each train line using `UI.println(line)`
- List the stations along a given train line by writing a `listStationsOnLine` method that **only** prints each station using `UI.println(station)`



The screenshot shows a Java Swing application window with a menu bar containing "All Stations" and "All Lines". Below the menu are four text input fields labeled "Station", "Train Line", "Destination", and "Time (24hr)". A "Lines of Station" button is at the bottom left. The main text area displays the output of the application:

```
Train lines for Johnsonville station:  
-----  
Wellington_Johnsonville (9 stations, 45 services)  
Johnsonville_Wellington (9 stations, 45 services)
```

Assignment 2a - Core

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment2PartA

- List the train lines that go through a given station by writing a listLinesOfStation method that only prints each train line using UI.println(line)
- List the stations along a given train line by writing a listStationsOnLine method that only prints each station using UI.println(station)

Train lines for Johnsonville station:

```

-----
Wellington_Johnsonville (9 stations, 45 services)
Johnsonville_Wellington (9 stations, 45 services)

```

45 services

Johnsonville_Wellington-services.data - Notepad

```

File Edit Format View Help
600 602 605 607 608 610 612 614 623
630 632 635 637 638 640 642 644 653
645 647 650 652 653 655 657 659 708
700 702 705 707 708 710 712 714 723
715 717 720 722 723 725 727 729 738
730 732 735 737 738 740 742 744 753
745 747 750 752 753 755 757 759 808
800 802 805 807 808 810 812 814 823
815 817 820 822 823 825 827 829 838
830 832 835 837 838 840 842 844 853
845 847 850 852 853 855 857 859 908
900 902 905 907 908 910 912 914 923
930 932 935 937 938 940 942 944 953
1000 1002 1005 1007 1008 1010 1012 1014 1023
1030 1032 1035 1037 1038 1040 1042 1044 1053
1100 1102 1105 1107 1108 1110 1112 1114 1123
1130 1132 1135 1137 1138 1140 1142 1144 1153
1200 1202 1205 1207 1208 1210 1212 1214 1223
1230 1232 1235 1237 1238 1240 1242 1244 1253
1300 1302 1305 1307 1308 1310 1312 1314 1323
1330 1332 1335 1337 1338 1340 1342 1344 1353
1400 1402 1405 1407 1408 1410 1412 1414 1423
1430 1432 1435 1437 1438 1440 1442 1444 1453
1500 1502 1505 1507 1508 1510 1512 1514 1523
1530 1532 1535 1537 1538 1540 1542 1544 1558
1600 1602 1605 1607 1608 1610 1612 1614 1628
1615 1617 1620 1622 1623 1625 1627 1629 1643
1630 1632 1635 1637 1638 1640 1642 1644 1658
1645 1647 1650 1652 1653 1655 1657 1659 1713
1700 1702 1705 1707 1708 1710 1712 1714 1728
1715 1717 1720 1722 1723 1725 1727 1729 1743
1730 1732 1735 1737 1738 1740 1742 1744 1758
1745 1747 1750 1752 1753 1755 1757 1759 1813
1800 1802 1805 1807 1808 1810 1812 1814 1828
1815 1817 1820 1822 1823 1825 1827 1829 1843
1830 1832 1835 1837 1838 1840 1842 1844 1853
1845 1847 1850 1852 1853 1855 1857 1859 1913
1900 1902 1905 1907 1908 1910 1912 1914 1923
1930 1932 1935 1937 1938 1940 1942 1944 1953
2000 2002 2005 2007 2008 2010 2012 2014 2023
2030 2032 2035 2037 2038 2040 2042 2044 2053
2100 2102 2105 2107 2108 2110 2112 2114 2123
2130 2132 2135 2137 2138 2140 2142 2144 2153
2230 2232 2235 2237 2238 2240 2242 2244 2253
2330 2332 2335 2337 2338 2340 2342 2344 2353

```

Johnsonville_Wellington-stations.data - Notepad

```

File Edit Format View Help
Johnsonville
Raroa
Khandallah
Box-Hill
Simla-Crescent
Awarua-Street
Ngaio
Crofton-Downs
Wellington

```