

COMP101 Lab 6: Fourth Assessed Coursework

Tunnel Toll Charges

worth 13% of the final mark
assignment four of seven

Failure to submit this assignment or submission of work that is deemed not to be a reasonable attempt will result in the failure of the module. The completion of the implementation and report as described below will obtain 90% of the marks. To obtain the final 10% students should also complete the Extended Requirements. This may involve doing some additional reading beyond what has been presented in lectures or more complex programming. Only attempt the extended requirements if you are confident with programming.

Learning Outcomes. This addresses the following learning outcomes, to

- be able to implement, compile, test and run Java programmes, comprising more than one class, to address a particular software problem;
- understand how to include arithmetic operators and constants in a Java program;
- demonstrate the ability to employ various types of selection constructs in a Java program.

Requirements. The cost to use a tunnel (such as the Mersey Tunnel) depends upon the type of vehicle and the time of day it is being used. The type of vehicle is classified by a number of factors, such as the number of wheels of the vehicle, the length, the number of axles, and/or the weight of the vehicle. The cost is also determined whether the tunnel is used during the week, or on the weekend.

The tunnel toll costs of these vehicles during the week, along with their classification, is summarized in the following table:

Number of wheels	Length	Number of axles	Weight (metric tons)	Class	Type	Cost (weekday)
< 4	-	-	-	Class 1	Motorbike	2.00
4	≤ 15ft	2	-	Class 2	Car	3.20
≥ 4	> 15ft	3-6	≤ 2	Class 3	Car with trailer	3.80
4	> 15ft	2	-	Class 4	Van	4.40
≥ 4	> 15ft	≥ 3	> 2 and ≤ 3.5	Class 5	Small lorry/Bus	8.00
≥ 4	> 15ft	> 6	> 3.5	Class 6	Large lorry	12.00

Note: The cost of the tunnel on the weekend is 3/4 of the cost for a weekday.

Design, implement and test a Java program which, given the number of wheels (an integer greater than or equal to 2), the length of the vehicle (a real number greater than 0), the number

of axles (an integer greater than or equal to 2), the weight of the vehicle in metric tons, and whether the tunnel is being used on a weekday or not (you can use a Boolean variable), determines and prints out the classification (1–6), the name of that class of vehicle, and the cost of using the tunnel.

Notes.

- As you can see, not all of the possible inputs are relevant for each vehicle classification. (For example, if the user inputs “2” for the number of wheels, there is no need to ask for the length, number of wheels, etc.)

Think carefully about a method to gather the input, and the order in which to collect it, to minimize the number of questions that you have to ask the user.

- The input value for the number of wheels should be an integer greater than or equal to 2. You should disallow values less than 2 as input.
- The input value for length (where appropriate) should be a real number greater than 0. You should disallow negative or zero values as input.
- The input value for the number of axles (where appropriate) should be an integer greater than or equal to 2. You should disallow values less than 2 as input.
- The weight (where appropriate) should be a real number greater than 0. You should disallow negative or zero values as input.
- You should provide pseudocode for (at least) your main method and the classification method.
- Split your code into self contained methods. Remember to use constants for fixed values.

Hint. Recall that a class can have several constructors, provided that the constructors differ in the number of parameters and/or the type of parameters given as input to the constructor.

Examples. Consider a vehicle with 3 wheels traveling on a weekday. This would be classified as *class 1 motorbike*, whatever the length, number of axles, and weight, and would be charged £2.00.

A vehicle with 8 wheels, length of 20ft, 6 axles, and weight of 3.1 metric tons would be classified *class 5 small lorry*. If it was traveling on the weekend it would be charged £6 ($=£8 \cdot (3/4)$).

The output from these examples is given below.

```
$ java TunnelTollChargeUser
Input the number of wheels (an integer greater than 0): 3
Input true if travelling on a weekday (Mon-Fri) and false otherwise: true

Wheels: 3, weekday
Vehicle class: 1
Vehicle type: Motorbike
Vehicle charge: 2.00
```

```
$ java TunnelTollChargeUser
Input the number of wheels (an integer greater than 0): 8
Input the length of the vehicle in feet (a double greater than 0.0): 20
Input the number of axles (an integer greater than 2): 6
Input the weight of the vehicle in metric tons: 3.1
Input true if travelling on a weekday (Mon-Fri) and false otherwise: false

Wheels: 8, length: 20, axles: 6, weight: 3.1, weekend
Vehicle class: 5
Vehicle type: Small lorry/Bus
Vehicle charge: 6.00
```

Extended Requirements. As well as paying for each journey, the tunnel offers a monthly pass for a single vehicle that can be used for any journey on weekdays and/or weekends. The cost of this is given below.

Vehicle Class	Pass Cost
1	70.00
2,3,4	130.00
5,6	350.00

Add methods that allow the input of the expected number of weekday and weekend journeys per month for some class of vehicle. Calculate and output the total cost of the expected journeys (weekday, weekend, and total) paying per journey. Compare this with the cost of a monthly pass and state whether it is cheaper to to buy a pass, it is the same, or it is not worth buying a pass.

Submission Instructions. Your submission, should consist of a report (a PDF file) and implementation files.

- The report (a PDF file) should consist of

Requirements: Summary of the above requirements statement.

Analysis and Design: A short (one paragraph) description of your analysis of the problem including a Class Diagram outlining the class structure for your proposed solution, and pseudocode for key methods. Note that your solution should comprise (at least) two classes: an “application class” and a “target class”.

Important: Make certain that your main application class (the class with the “main” method) is called “TunnelTollChargeUser”.

If you do the extended requirements, make certain that your main application class is called “TunnelTollChargeUserExt”.

Testing: A set of proposed test cases presented in tabular format including expected output, and evidence of the results of testing (the simplest way of doing this is to cut and paste the result of running your test cases into your report).

- The implementation should consist of

Your Java source files, i.e. the relevant .java files, not the class (.class) files.

Upload your files to

<https://sam.csc.liv.ac.uk/COMP/Submissions.pl>

(you will need to log in using your Computer Science username and password).

Submission Deadline. Monday 14th November, 5pm.

Mark Scheme Marks will be awarded for

- Analysis and Design 15%
- Implementation 65%
- Testing 10%
- Extended requirements 10%

Please see the module web page for the feedback form.

Note.

- Because submission is handled electronically, ANY FILE submitted past the deadline will be considered (at least) ONE DAY late. Late submissions are subject to the University Policy (see www.csc.liv.ac.uk/departments/regulations/practical.html).
- Please make sure your Java classes successfully compile and run on ANY departmental computer system, both Windows and Unix. For this reason the use of powerful IDEs like NetBeans is discouraged.
- Note this is an individual piece of work. Please note the University Guidelines on Academic Integrity (see https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf).