

CSE1100/CSE4100 Semester 1, 2013

Programming Assignment

Assessment: This programming assignment is worth 20% of the final mark for this unit.

Due Date: Monday May 20, 2013, at 9.30 am

Delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime.

This is an individual assignment. You are not permitted to work as a group when writing this assignment.

Copying, Plagiarism: Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. The Department of Computer Science and Computer Engineering treats academic misconduct seriously. When it is detected, penalties are strictly imposed. Refer to the unit guide for further information and strategies you can use to avoid a charge of academic misconduct. All submissions will be electronically checked for plagiarism.

Objectives: The general aims of this assignment are:

- To analyze a problem in an object-oriented manner, and then design and implement an object-oriented solution that conforms to given specifications
- To practise using inheritance in Java
- To practise file input and output in Java
- To make implementations more robust through mechanisms such as exception handling.

Submission Details and Marking Scheme: Instructions on how to submit electronic copies of your source code files from your latcs6 (or latcs5) account and a marking scheme overview are given at the end. If you have not been able to complete a program that compiles and executes containing all functionality, then you should submit a program that compiles and executes with as much functionality as you have completed. (You may comment out code that does not compile.)

NOTE: While you are free to develop the code for this assignment on any operating system, your solution must run on the latcs6 (or latcs5) system.

Problem Description

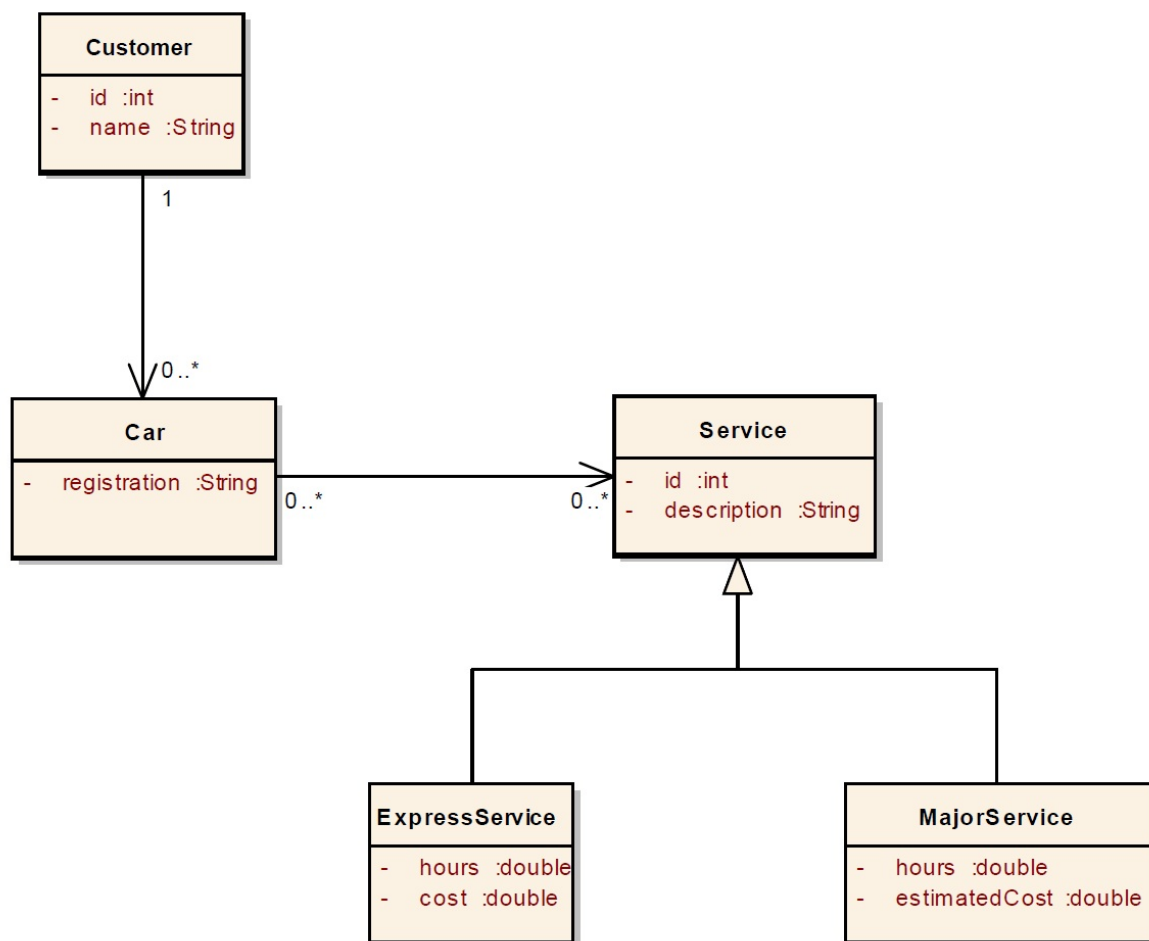
A car service centre needs a system to keep information about their customers and bookings about their cars.

Each customer has a unique ID and a name. Each customer can have more than one car registered in this system, and for each car the customer can book different services, such as express service, major service. Each booking is only for one of the two types of service. After the service, the staff will use the system to calculate the total cost and charge the customer.

An express service can be done within two hours. It is charged as follow: with an hour it is \$100, and over an hour it is \$150. This charge covers the labour cost and consumable materials, such as engine oil, etc.

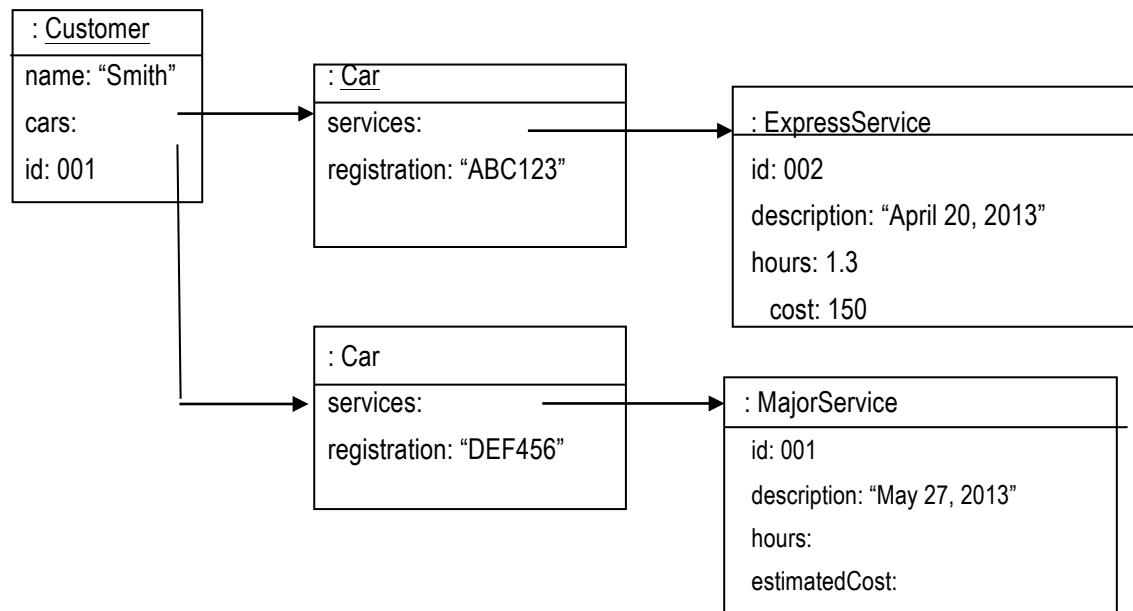
A major service is done according to the car manufacture's manual, the cost is based on the how many hours of the major service plus the consumable materials cost. The hourly labour cost is \$50. For example, a major serve of 6 hours plus \$100 material cost (the material cost will ben entered by staff when calculating the total cost) will cost $\$50 \times 6 + \$100 = \$400$.

A design has been made of the system, and is presented in the class diagram below.



A snapshot

A snapshot of a customer's data may look like what shown in the diagram below.



The car service centre staff can use the system to:

- Add a customer (enter details such as id, name.)
- Add a car (for a customer, enter car registration)
- Make a booking (for a customer's car, enter service date, service type)
- Modify a booking (change the service date and service type)
- Cancel a booking (of a customer)
- Calculate service cost (given the car registration and service id, calculate and display the cost)
- Retrieve a customer record by the id (including the customer's cars, bookings, and service charges)

Assume that we can take 100 to be the maximum number of customers, and each customer has maximum 10 cars. For each car the system will keep the recent 50 service records.

Task:

a. Write a program, called CarServiceBooking, that presents the following menu:

```
== Car Service Booking System ==  
1. Add a customer  
2. Add a car  
3. Make a booking  
4. Retrieve details of a booking  
5. Modify a booking  
6. Cancel a booking  
7. Calculate service cost  
8. Retrieve a customer record by the id  
9. Save all the data to a text file  
E. Exit  
Please enter an option (1-9 or E):
```

And the user can select the above options to call relevant functions.

b. Include enough exception handling features to make the menu program robust.

Additional task for CSE4IOO Students:

Implement a VIPCustomer class, subclass of Customer, and the service charge is discounted according to how many years the VIPCustomer is with the car service center, for example 5 years 5% off, 10 years 10%. Add an item in the menu to add VIP customers.

Electronic Submission of the Source Code

- Submit all the Java files that you have developed in the tasks above.
- The code has to run under Unix on the latsc6 (or latsc5) machine.
- You submit your files from your latsc6 (or latsc5) account. Make sure you are in the same directory as the files you are submitting.
- Submit each file separately using the submit command. For example

```
submit IOO Customer.java
```

```
.....
```

```
submit IOO CarServiceBooking.java
```

- After submitting the files, you can run the following command that lists the files submitted from your account:

```
verify
```

- You can submit the same filename as many times as you like before the assignment deadline; the previously submitted copy will be replaced by the latest one.

Marking Scheme Overview

You are required to attend a marking session to demonstrate your assignment (in your allocated lab session in Week 12). Your understanding of your assignment may be used as a weighting factor on the final assignment mark.

- Implementation (Execution of code): 90% (Do all parts of the programs execute correctly? Note your programs must compile and run to carry out this implementation marking.)
- Program Design and Structure, and Code Quality: 10% (Does the program conform to specifications?

Does the program solve the problem in a well-designed manner? Does the program follow good programming practices? Does the indentation and code layout follow a good, consistent standard? Are the identifiers meaningful? Note (Javadoc) comments are NOT required.)

Return of Assignments

Department policy requires that assignments are returned within 3 weeks of the submission date. Students will be notified by email and via the CSE1100 website news when marking sheets are available for collection from the department's general office.