

CE00527-2 Further Object Oriented Programming Tutorial 3

In this tutorial you will investigate using inner classes, and start building a Graphical User Interface for the Showroom system.

Part 1 - Using Inner Classes - Basic Exercise

1. Implement the Student (with inner class Address) class given in Lecture 2 Week 2. Test it by implementing a TestStudent class which contains the main method shown on p. 18, Lecture 2, Week 2.

Add a toString() method to Student which returns the Student name, and their home and uni addresses (if not null). Add a line to TestStudent which calls the toString() method of the student object s1 and prints out the result.

What is the relationship of the address object anotherAddress to the Student object s1?

Add a method setUniAddress() to the Student class so a uni address can be assigned to a Student object. Modify TestStudent so the uni address "72 Nottingham Drive" is assigned to s1 before the toString method of s1 is called and printed out.

Put the code listing for the Student class, TestStudent class, and their output, in your portfolio.

Now modify Address so it is a **private** inner class of Student. Which of the statements in TestStudent are no longer valid, and why? You should still be able to assign a uni address to a Student object – if your original method no longer works, write an overloaded method that does.

Modify Address again so it is a **public static** inner class of Student. Which method of Address is no longer valid and why? Modify the offending method so it compiles. Update TestStudent so it creates and tests the Address object anotherObject using this static inner class.

Put the answers to the above questions, and description of any changes to code listings and output (include the relevant code and output) in your portfolio.

Tutorial continues on next page...

Part 2 - Designing a user interface for the Showroom system -Advanced Exercise

2. In tutorial 1 you designed and built classes to hold information about the Vehicles in a Showroom. Design a GUI to browse backwards and forwards through the Vehicles. Include three buttons:
 - previous vehicle
 - next vehicle
 - sellJust the details of the current Vehicle (manufacturer, model, VIN, manufacturer date, cost, Taxband and CO2, sold (Boolean), sold date, customer) should be displayed at any point in time. You should label the GUI components (JButtons, JPanels, etc) and layout managers used. The sell button invoke the sell() method of the current Vehicle. Use a dialog box to capture the Customer name, and use today's date as the date sold.

Part 3 - Implementing the user interface-Advanced Exercise

3. Now build and test the user interface. Use swing components where appropriate, and anonymous inner classes for the event listeners. Make full use of the Vehicle and Showroom classes. Try to keep the user interface as separate from the Showroom class as possible; it would make sense to start with a frame derived class that "has an" instance of the Showroom class. Any access to Vehicle objects should be made via the instance of the Showroom class.

You should put in your portfolio:

- **Student/Address and TestAddress code listings, output, and answers to the questions of Part 1**
- **your design for the Showroom system user interface. Be sure to label the GUI components and layout managers used**
- **a code listing for your Showroom system user interface**
- **a screen shot of the running GUI program**