also globally visible so no identifier is needed. Bike has an array of identifiers for all the :Bikes. When it finds the :Bike with a matching bike#, it returns its object identifier to :IssueBikeUI which stores the identifier. The :IssueBikeUI then calls the matching :Bike to get the details.

Step 3 :IssueBikeUI uses the :Bike identifier stored in step 2 to send it a calculateCost(no.Days) message. The number of days, used as a parameter to this message, is input by the actor and stored by :IssueBikeUI for use in step 6.

Step 4 :IssueBikeUI creates a new Customer object, passing in the name, postcode and telephone details. It then stores the new :Customer identifier.

Step 5 :IssueBikeUI creates a new Payment object, passing the stored :Customer identifier from step 4. :IssueBikeUI stores the object identifier of the new Payment object.

Step 6 :IssueBikeUI creates a new Hire object. The attribute startDate is set to today's date, the attribute, numberOfDays, is set to the number of days stored by :IssueBikeUI in step 3.

Step 7 :IssueBikeUI sends a calcTotalPayment() message to :Payment, using the identifier it stored in step 5.

Step 8 :Payment issues a receipt. To do this it needs to get information from Customer object (name and postcode); the relevant :Customer's identifier was passed as an argument to its constructor.

Chapter 11

Exercise 11.1

Part a The constructors are on lines:

04 public European()
14 public Briton()
25 public Frenchman()
36 public German()
47 public Italian().

Part b The keyword is: extends, on line 11.