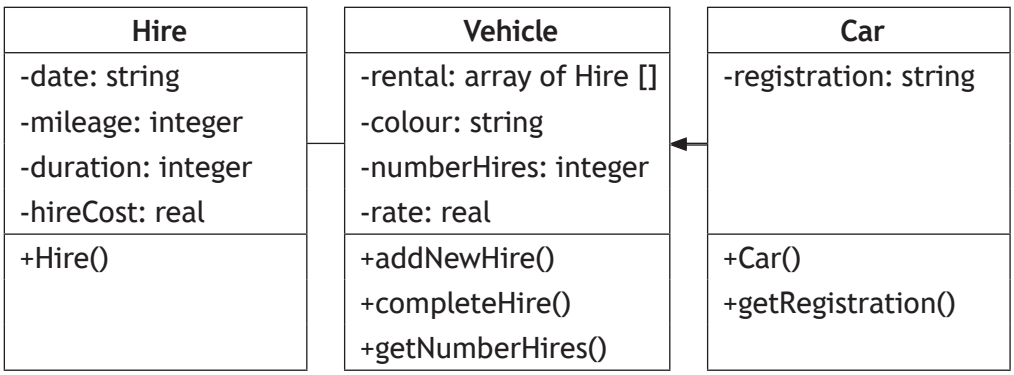
3. GlasgowGreen uses an object-oriented program to track the use of hire vehicles across the city.

A simplified UML class diagram for the program is shown below.



Some of the object-oriented code used in the program is shown below.

```
Line 1    CLASS Vehicle IS { ARRAY OF Hire rental,  STRING colour,
          INTEGER numberHires, REAL rate }
Line 2      METHODS
Line 3        PROCEDURE addNewHire(STRING date)
Line 4           SET THIS.rental[THIS.numberHires].date TO date
Line 5           SET THIS.rental[THIS.numberHires].mileage TO 0
Line 6           SET THIS.rental[THIS.numberHires].duration TO 0
Line 7           SET THIS.rental[THIS.numberHires].hireCost TO 0.0
Line 8        END PROCEDURE

Line 9        PROCEDURE completeHire (INTEGER mileage, INTEGER
              duration)
Line 10          SET THIS.rental[THIS.numberHires].duration TO
                 duration
Line 11          SET THIS.rental[THIS.numberHires].mileage TO
                 mileage
Line 12          SET THIS.rental[THIS.numberHires].hireCost TO
                 THIS.rate * THIS.rental[THIS.numberHires].mileage
Line 13          SET THIS.numberHires TO THIS.numberHires + 1
Line 14       END PROCEDURE
          …
Line 23   END CLASS
```

3. (continued)

```
Line 24  CLASS Car INHERITS Vehicle WITH { STRING registration }
Line 25    METHODS
Line 26      CONSTRUCTOR ( STRING registration, STRING colour )
Line 27         DECLARE THIS.registration INITIALLY registration
Line 28         DECLARE THIS.rate INITIALLY 7.25
Line 29         DECLARE THIS.rental AS ARRAY OF Hire INITIALLY[]
Line 30         DECLARE THIS.colour INITIALLY colour
Line 31         DECLARE THIS.numberHires INITIALLY 0
Line 32      END CONSTRUCTOR
         ...
```

A `Car` object is instantiated using the following statement.

```
DECLARE car1 AS Car INITIALLY Car ("ABC123", "Red")
```

(a) State what is stored in all of the instance variables of the `car1` object upon instantiation. **1**

(b) Using appropriate object-oriented terminology, explain why the following statement would be invalid. **2**

```
SET car1.registration TO "DEF456"
```

(c) Vehicle hire across the city has proved to be very popular and GlasgowGreen has decided to expand the business by introducing pedal bikes for hire.

A second subclass of the `Vehicle` class called `Bike` must be added to the program.

(i) Explain why the use of subclasses with inheritance will reduce the time needed to add the `Bike` class to the program. **1**

(ii) The `Bike` class will have one additional private property, `bikeID`, which is a string value, and three additional public methods called `Bike()`, `completeHire()` and `getBikeID()`.

Complete the UML class diagram to include the `Bike` class, showing all properties and methods. You need only display the name of each class in the existing UML class diagram. **2**

[Turn over

MARKS

3. (c) (continued)

(iii) The variable `vehicleArray` is an array of `Vehicle` objects. This array stores the details of all the vehicles owned by GlasgowGreen.

An object of the `Car` class is stored in the `vehicleArray` in position 20.

Using appropriate object-oriented terminology, explain why the following codes returns an error.

```
SET carReg TO vehicleArray[20].getRegistration()
```

3

(d) The variable `bikeArray` is an array of `Bike` objects. This array stores the details of the 125 bikes owned by GlasgowGreen.

GlasgowGreen wants to know the ID of the bike that has been hired most often.

Using a programming language of your choice, write the code to find and display the ID of the bike that has been hired most often. **3**