1 (a) network ID: the ID common to all devices on a network
host ID: the unique ID of a particular device on a network

(b) 205 = 11001101. It starts with 110, so it is a Class C address

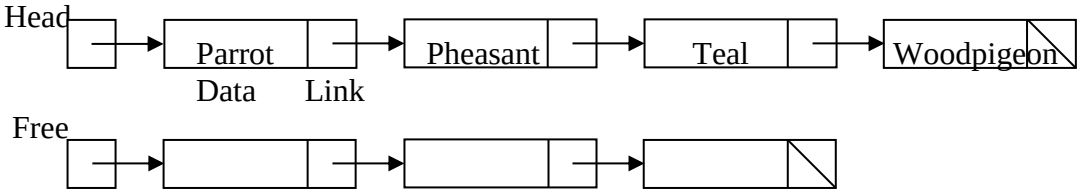
(c) network ID: 205.123.4
host ID: 192

2 (a)

(i) E – A two dimensional array because the elements of the array form a table
(ii) D – A hash table because each customer ID could be stored alongside its hashed value
(iii) B – A queue because the traffic would form a queue along the street (leaving from one end, joining from the other)
(iv) A – A stack because the web pages would be retrieved in reverse order being popped from a stack
(v) C – A binary tree because data could be searched from it using binary search
(vi) F – A linked list because the records could be inserted into their correct position without having to sort the file into order

(b) (i)

- Head of list table . . .
- with entry for this list
- pointers used
- with sensible null pointer for last item
- Attempt to represent free space list



(ii)

- Traverse to the required node (Pheasant) from Head pointer
- Link of predecessor of deleted node (Parrot) points to successor of deleted node (Teal)
- Link of deleted node (Pheasant) points to 1st free node
- Header of free list now points to the deleted node (Pheasant)

(iii)

```
PROCEDURE Reverse (Current)
        Next ← Current.Link
        IF Next <> NULL
                Reverse (Next)
        ENDIF

        OUPUT Current.Data
ENDPROCEDURE
```

3

(a) Protocol is a set of pre-agreed rules to govern communication (between devices)

Protocol is necessary as there are different devices and different networks and therefore there need to be rules that specify how these differences are to be resolved, if communication is to be possible.

(b) [Any 3]

Packet Switching	**Circuit Switching**
has no established route	establishes a route along which to send packets
packets being sent on individual routes	has packets all on same route
message cannot be (easily) intercepted	message can because all on same route
packets need to be reordered	packets remain in correct order
maximises use of network	ties up large areas of network

4

(a) An owner may own more than one vehicle or an owner may bring different vehicles to the service center or a vehicle might be owned by more than one owner at different times.

Making it easier to transfer a vehicle from one owner to another.

Minimise data duplication or eliminate data redundancy and data inconsistency

Eliminate update and insertion anomalies.

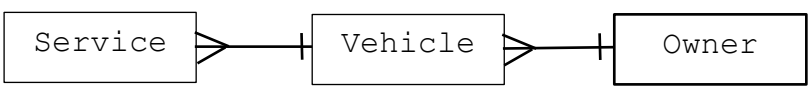
(b)

Service(ServiceID, VehicleRegNo, ServiceDate, ServiceDuration)

Vehicle(VehicleRegNo, Brand, Model, OwnerID)

Owner(OwnerID, OwnerName, OwnerContactNo)

(c)



(d)

VehicleRegNo and ServiceDate

(e)

```
SELECT
Service.ServiceID,
Vehicle.VehicleRegNo,
Service.ServiceDate,
Vehicle.Brand,
Vehicle.Model,
Owner.OwnerName,
Owner.OwnerContactNo
FROM Vehicle INNER JOIN Service on Vehicle.VehicleRegNo =
Service.VehicleRegNo
INNER JOIN Owner on Vehicle.OwnerID = Owner.OwnerID
WHERE Service.ServiceDuration=0
```

(f)

[4M] 3 classes – 1 superclass, 2 subclasses with private attributes and public get/set methods for the attributes. Arrow from subclass to superclass to show

[1M] – Inheritance - Arrow from subclass to superclass to show inheritance

[1M] – Polymorphism – show methods for all 3 classes

(g) The sub-classes (Non-ElectricVehicleService and ElectricVehicleService) inherit all the attributes and methods of the Service class.

The methods getVehicleRegNo**,** setVehicleRegNo**,** getServiceDate**,** setServiceDate**,** setServiceDuration**,** getServiceDuration, setOwnerName, getOwnerName **…** are inherited without changes to the implementation and the no coding are required, hence code reused is achieved.

(h) Polymorphism refers to an object's ability to take different forms. It allows subclasses to have methods with the same name as methods in their superclasses. It gives the ability for a program to call the correct method depending on the type of object that is used to call it.

The method showInfo() in both the subclasses **overrides** the superclass showinfo(). If the subclass object (Non-ElectricVehicleService) is used to call showinfo() then the subclass's version of the method is invoked. If the superclass object (Service) is used to call *show*, then the superclass method will be invoked.