

1 The composite record data type, `ClubMember`, is defined in pseudocode as:

```

TYPE ClubMember
  DECLARE Code : INTEGER
  DECLARE LastName : STRING
  DECLARE FirstName : STRING
  DECLARE Telephone : STRING
  DECLARE JoinDate : DATE
  DECLARE Fees : REAL
  DECLARE FeesPaid : BOOLEAN
ENDTYPE

```

(a) (i) Write the **pseudocode** statement to set up a variable for one record of the composite data type, `ClubMember`.

.....
 [1]

(ii) Write the **pseudocode** statements to assign the following values to the variable set up in part (a)(i):

- 984632 to `Code`
- TRUE to `FeesPaid`

.....

 [2]

(b) An enumerated data type, `Activity`, is required, so that a new field, `Choice`, can be added to the composite data type, `ClubMember`, to allow members to choose an activity.

(i) Write the **pseudocode** statement for the type declaration of `Activity` to hold the names of the available activities:

Badminton, Football, Golf, Snooker, Swimming, Tennis.

.....

 [2]

(ii) Write the **new pseudocode** statement required to update the declaration of `Choice` in the definition of `ClubMember`.

.....

 (1) [1]

1 (a) An enumerated data type, `Spectrum`, is required to hold the names of colours.

(i) Write the **pseudocode** statement for the type declaration of `Spectrum` to hold the names of the colours available:

Red, Orange, Yellow, Green, Blue, Indigo, Violet.

.....
 [2]

(ii) Identify **two** characteristics of the list of values given in an enumerated data type declaration.

1

 2
 [2]

(b) `ColourData` is a composite data type to store definitions of colours.

Write **pseudocode** statements to declare `ColourData` to hold the following fields:

Field	Example data
ColourCode	XYZ12345
Colour	Red
Wavelength	650
Frequency	4.62
PrimaryColour	Yes

Use the most appropriate data type in each case, including the enumerated data type `Spectrum` from part a(i).

.....

 (2) [4]

2 A program stores data about trains and train stations using Object-Oriented Programming (OOP).

The class `Train` stores the data about the trains:

Train	
<code>TrainIDNumber : String</code>	stores the train ID number
<code>Route : Integer</code>	stores the route number the train is travelling
<code>Constructor()</code>	initialises <code>TrainIDNumber</code> and <code>Route</code> to the parameter values
<code>GetTrainIDNumber()</code>	returns the train ID number
<code>GetRoute()</code>	returns the route number the train is travelling

(a) (i) Write program code to declare the class `Train` and its constructor.

Do **not** declare the other methods.

All attributes should be private.

Use your programming language appropriate constructor.

If you are writing in Python, include attribute declarations using comments.

Save your program as **Question2_N25**.

Copy and paste the program code into part **2(a)(i)** in the evidence document.

[4]

(ii) The methods `GetTrainIDNumber()` and `GetRoute()` return the appropriate attribute.

Write program code for `GetTrainIDNumber()` and `GetRoute()`

Save your program.

Copy and paste the program code into part **2(a)(ii)** in the evidence document.

[3]

(b) The program is tested with four trains:

train ID number	route
12ADV	134
33ART	20
9FKF	3
21VBC	24

Write program code to declare an instance of `Train` for each of the **four** trains.

Save your program.

Copy and paste the program code into part **2(b)** in the evidence document.

[2]

(c) The class `Station` stores the data about the stations:

Station	
<code>StationID : String</code>	stores the station ID
<code>NumberPlatforms : Integer</code>	stores the number of platforms at the station
<code>Trains[0:9] : Train</code>	stores the trains currently at the station platforms
<code>NumberTrains : Integer</code>	stores the number of trains currently at the station platforms
<code>Constructor()</code>	initialises <code>StationID</code> and <code>NumberPlatforms</code> to the parameter values, initialises <code>Trains</code> to an empty array and <code>NumberTrains</code> to 0
<code>GetTrains()</code>	returns a string containing data about the trains currently at the station platforms
<code>AddTrain()</code>	takes a <code>Train</code> parameter and stores it if there is a platform available; each platform can only have one train

(i) Write program code to declare the class `Station` and its constructor.

Do **not** declare the other methods.

All attributes should be private.

Use your programming language appropriate constructor.

If you are writing in Python, include attribute declarations using comments.

Save your program.

Copy and paste the program code into part 2(c)(i) in the evidence document.

[3]

(ii) The method `AddTrain()` takes a `Train` parameter. The method compares the attributes `NumberTrains` and `NumberPlatforms` to identify if there is a platform available (a platform currently with no train).

The method returns `FALSE` if there are no platforms available.

If there is a platform available, the method:

- stores the `Train` parameter in the array `Trains`
- updates the appropriate attribute(s)
- returns `TRUE`

Write program code for `AddTrain()`

Save your program.

Copy and paste the program code into part 2(c)(ii) in the evidence document.

[4]

(iii) The method `GetTrains()` returns the string "There are no trains" if there are no trains at the station platforms.

If there are trains at the station platforms, the method returns a string in the format:

The trains at station <StationID> are:
<TrainIDNumber> on route number <Route>

The line <TrainIDNumber> on route number <Route> is repeated for each train at the station platforms.

For example: If the station with the station ID "NT1" has two trains with the train ID numbers "48RTG", "6UFH", the method will produce this output:

The trains at station NT1 are:
48RTG on route number 43
6UFH on route number 12

Write program code for `GetTrains()`

Save your program.

Copy and paste the program code into part 2(c)(iii) in the evidence document.

[6]

(d) The program is tested with two stations:

station ID	number of platforms
STH	2
NTH	1

(i) Write program code to amend the main program to declare an instance of `Station` for each of the **two** stations.

Save your program.

Copy and paste the program code into part **2(d)(i)** in the evidence document.

[2]

(ii) The four trains attempt to stop at the following stations in the order given:

- Train 12ADV, station STH
- Train 33ART, station STH
- Train 9FKF, station STH
- Train 21VBC, station NTH

Write program code to amend the main program to:

- add each train to the given station using `AddTrain()`
- output "Station is full" for any train where the return value indicates it cannot be added to the station
- output the trains at each station using `GetTrains()`

Save your program.

Copy and paste the program code into part **2(d)(ii)** in the evidence document.

[4]

(iii) Test your program.

Take a screenshot of the output(s).

Save your program.

Copy and paste the screenshot(s) into part **2(d)(iii)** in the evidence document.

[2]

1 A programmer is writing a program to manage bookings for a small taxi company. The programmer requires some user-defined data types.

(a) Write a **pseudocode** statement to declare the enumerated data type, `Vehicle`, to hold the identity code of each of the company's taxis:

M100, M230, T101, T102, T120, T150

.....
..... [2]

(b) Write **pseudocode** statements to declare the composite data type, `Booking`, to hold data about taxi bookings. The data required includes:

- booking number (any combination of letters and numbers)
- destination
- client name
- client telephone number
- date of departure
- address for pick-up
- the identity code of the taxi used.

Use the most appropriate data type in each case, including the enumerated data type from part (a).

.....
.....
.....
.....
.....
.....
.....
.....
.....
..... [4]

1 Data types can be defined using pseudocode.

The composite record data type, `Departure`, is used to represent flights from Cambridge Airport and is defined in pseudocode as:

```
TYPE Departure
    DECLARE FlightNumber : STRING
    DECLARE Destination : STRING
    DECLARE FlightDate : DATE
    DECLARE Gate : STRING
    DECLARE Airline : STRING
ENDTYPE
```

A variable, `Flight1`, is declared in pseudocode as:

```
DECLARE Flight1 : Departure
```

(a) Write **pseudocode** to store the following details to `Flight1`:

Field	Data
FlightNumber	SB2789
Destination	Dublin
FlightDate	30/07/2025
Gate	N03
Airline	Cambridge Airways

.....

.....

.....

.....

.....

.....

.....

..... [3]

(b) The data type for `Gate` is changed to an enumerated data type, `GateID`.

(i) Write a **pseudocode** statement to declare `GateID` to hold the identity codes for the airport gates:

N01, N02, N03, W01, W02, W03, W04

.....

..... [2]

(ii) Write the new **pseudocode** statement required to replace the declaration of `Gate` in `Departure`.

1 A programmer is writing a program to manage a video library. They require a user-defined data type.

(a) Write **pseudocode** statements to declare the composite data type `VideoLibrary` to hold data about each video in the collection. This data includes:

- identity code (any combination of letters and numbers)
- title
- year released
- date purchased
- format (for example DVD, Blu-ray, 4K, MP4)
- running time (minutes)

Use the most appropriate data type in each case.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) Identify **one** field in `VideoLibrary` that could be an efficient enumerated data type and give a reason for your choice.

Field

.....

Reason

.....

- 9 A veterinary surgery wants to create a class for individual pets. Some of the attributes required in the class are listed in the table.

Attribute	Data type	Description
PetID	STRING	unique ID assigned at registration
PetType	STRING	type of pet assigned at registration
OwnerTelephone	STRING	telephone number of owner assigned at registration
DateRegistered	DATE	date of registration

- (a) State **one** reason why the attributes would be declared as `PRIVATE`.

.....
 [1]

- (b) Complete the class diagram for `Pet`, to include:

- an attribute and data type for the name of the pet
- an attribute and data type for the name of the owner
- a method to create a `Pet` object and set attributes at the time of registration
- a method to assign a pet ID
- a method to assign the date of registration
- a method to return the pet name
- a method to return the owner's telephone number.

Pet	
PetID	: STRING
PetType	: STRING
OwnerTelephone	: STRING
DateRegistered	: DATE
.....	:
.....	:
.....	
.....	
.....	
.....	

- 3 (a) Describe the user-defined data type **record**.

.....

 [3]

- (b) A programmer defines a record, `Order`, to store the following data:

- account number
- order number
- order price
- order date.

Write **pseudocode** statements to define this record.

.....

 [4]

6 (a) Describe the user-defined data type **set**.

.....

.....

.....

.....

.....

..... [3]

(b) Write **pseudocode** statements to declare the set data type, `SymbolSet`, to hold the following set of mathematical operators, using the variable `operators`.

+ - * / ^

.....

.....

.....

.....

.....

.....

.....

..... [4]