

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]