2 A computer game is being developed. The game has 10 different characters that are all active in the game.

Part of the game is being written using object-oriented programming.

The class `Character` stores data about the characters. Each character has a name and the x coordinate and y coordinate of their current position.

Character	
`Name : STRING`	stores the name of the character
`XCoordinate : INTEGER`	stores the x coordinate
`YCoordinate : INTEGER`	stores the y coordinate
`Constructor()`	initialises `Name`, `XCoordinate` and `YCoordinate` from the values passed as parameters
`GetName()`	returns the name of the character
`GetX()`	returns the x coordinate of the character
`GetY()`	returns the y coordinate of the character
`ChangePosition()`	takes `XChange` as an integer parameter and adds it to the x coordinate takes `YChange` as an integer parameter and adds it to the y coordinate

(a) Write program code to declare the class `Character` and its constructor. Do **not** write program code for the other methods.

Use your programming language appropriate constructor.

All attributes must be private. If you are writing in Python, include attribute declarations using comments.

Save your program as **Question2_N22**.

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

[4]

(b) Write program code for the **three** get methods for the class `Character`.

Save your program.

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

[3]

(c) Write program code for the method `ChangePosition()`.

Save your program.

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

[2]

(d) The main program has a 1D array of characters. Each character is stored as an object of type `Character`.

The game has a maximum of 10 characters. The character names, x coordinates and y coordinates are stored in the file `Characters.txt` in the order:

- name
- x coordinate
- y coordinate.

For example, the first character in the file is named Amal, with the x coordinate 0 and the y coordinate 2.

Amend the main program by writing program code to:

- declare the array
- read in all 10 characters from `Characters.txt`
- store each character as an object in the array.

Save your program.

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

[7]

(e) The main program needs to read in a character's name from the user, search for the character in the array and store the index of its position. It repeats until the user enters a name that exists in the array.

Amend the main program by writing program code to perform this task.

Save your program.

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

[5]

(f) The user will enter a letter to identify the direction the chosen character from **part 2(e)** should move.

- If 'A' is input, the character moves left (x coordinate minus 1).
- If 'W' is input, the character moves up (y coordinate plus 1).
- If 'S' is input, the character moves down (y coordinate minus 1).
- If 'D' is input, the character moves right (x coordinate plus 1).

Amend the main program by writing program code to:

- take a letter as input until it is a valid move (A, W, S or D)
- change the position of the character using the appropriate method.

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

[7]

(g) (i) When a change to a character's position has been made, the program needs to output the character's name and the new x and y coordinates of the character, in the format:

```
Qui has changed coordinates to X = 83 and Y = 0
```

Amend the main program by writing program code to perform these tasks.

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

[2]

(ii) Test your program by inputting the following **four** items of data in the order given:

```
THOMAS
qui
X
A
```

Take a screenshot of the output.

Copy and paste the screenshot into **part 2(g)(ii)** in the evidence document.

[1]