6 A programmer wants to create a computer simulation of animals searching for food in a desert. The desert is represented by a 40 by 40 grid. Each position in the grid is represented by a pair of coordinates. `'A'` represents an animal and `'F'` represents food. At the start of the simulation, the grid contains 5 animals and 1 food source.

The following is an example of part of the grid.

	0	1	2	3	4	...	37	38	39
0	A					..			
1			F			..			
2						..		A	
3				A		..			
...	..	..	..	..	..	..	..	..	..
38				A		..	A		
39						..			

A timer is used. In each time interval, each animal randomly moves 0 or 1 position in a random direction. The program generates this movement by computing two random numbers, each of which can be –1, 0 or 1. The program adds the first random number to the across number and the second random number to the down number representing the animal's position.

For example:

- if 0 and 1 are generated, the across value does not change, the down value increases by 1
- if –1 and 1 are generated, the across value decreases by 1, and the down value increases by 1.

Each animal has an individual score. If the animal moves to a position in the grid with food (`'F'`):

- the animal's score increases by 1
- the food disappears
- one new animal (`'A'`) is randomly generated and added to the grid (to a maximum of 20 animals)
- one new food (`'F'`) is randomly generated and added to the grid.

The simulation is to be implemented using object-oriented programming.

The programmer has designed two classes, `Desert` and `Animal`.

The `Desert` class consists of:

- attributes
  - ○ `Grid`
  - ○ `StepCounter`
  - ○ `AnimalList`
  - ○ `NumberOfAnimals`
- methods
  - ○ `Constructor`
  - ○ `IncrementStepCounter`
  - ○ `GenerateFood`
  - ○ `DisplayGrid`

Each attribute consists of a value and a get and set method that allow access to the attributes.

The following table describes the attributes and methods for the `Animal` class.

Identifier	Data type	Description
`Constructor()`		Instantiate an object of the `Animal` class • Generate a pair of random numbers between 0 and 39. • Place animal at that random position. • Initialise the animal's score to 0.
`EatFood()`		• Delete the food. • Increase the score of the animal that called the method. • Call the `GenerateFood` method of the `Desert` class. • Call the `Constructor` method of the `Animal` class.
`Move()`		• Call the `GenerateChangeInCoordinate` method for each coordinate (across or down number) of the animal's position. • Moves the animal to the new space. • If there is food in the new position, call the `EatFood` method.
`Score`	`INTEGER`	Initialised to 0
`Across`	`INTEGER`	The across value, between 0 and 39
`Down`	`INTEGER`	The down value, between 0 and 39

(a) Write **program code** to declare the attributes and constructor for the `Animal` class.

You only need to write the set and get methods for the attribute `Across`.

You should also write:

- the constructor for the class
- set and get methods for the `Across` attribute only.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[6]

(b) The `Constructor` method of the `Desert` class:

- initialises an empty grid
- creates 5 animal objects which are added to the `AnimalList` (an array of animal objects currently on the grid)
- generates one food
- sets the `StepCounter` to 0.

Write **program code** for the `Constructor` method.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(c) **(i)** The function `GenerateChangeInCoordinate`:

- receives a coordinate (across or down number) as a parameter
- checks whether the coordinate's value is at a boundary of the grid
- returns a random change (–1, 0 or 1) that will keep the animal's position within the grid.

Write **program code** for the `GenerateChangeInCoordinate` function.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....[4]

(ii) The `Move` method uses the `GenerateChangeInCoordinate` function to calculate the new `Across` and `Down` values for an animal. If there is food in the new position in the grid, the animal eats the food.

Write **program code** for the `Move` method.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....[4]

(d) The programmer plans to add a graphic display to the program. The programmer will make use of a program library.

Explain what is meant by a program library.

....

....

....

....[2]