

3 A program stores data about animals using Object-Oriented Programming (OOP).

The class `Animal` stores the data about animals.

| Animal | |
|-------------------------------------|---|
| <code>Name : String</code> | stores the name of the animal |
| <code>Sound : String</code> | stores the sound the animal makes |
| <code>Size : Integer</code> | stores the size of the animal as an integer between 1 (smallest) and 10 (largest) |
| <code>Intelligence : Integer</code> | stores the intelligence of the animal as an integer between 1 (least) and 10 (most) |
| <code>Constructor()</code> | initialises all attributes to its parameter values |
| <code>Description()</code> | returns a string message that contains the data from the attributes |

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

Do **not** declare the other methods.

Use your programming language appropriate constructor.

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

Save your program as **Question3_J25**.

Copy and paste the program code into part **3(a)(i)** in the evidence document.

[4]

(ii) The method `Description()` creates and returns a string of the animal's data in the format:

```
"The animal's name is " <Name> ", it makes a " <Sound> ", its size is " <Size> " and its intelligence level is " <Intelligence>
```

For example:

```
The animal's name is Teddy, it makes a Bark, its size is 4 and its intelligence level is 6
```

Write program code for `Description()`

Save your program.

Copy and paste the program code into part **3(a)(ii)** in the evidence document.

[3]

(b) The class `Parrot` inherits from the class `Animal`

`Parrot` inherits the attributes from `Animal` and overrides the `Description()` method. The additional attributes and methods in the class `Parrot` are:

| Parrot | |
|------------------------------------|---|
| <code>WingSpan : Integer</code> | the width of the parrot's wings to the nearest cm |
| <code>NumberWords : Integer</code> | the number of words the parrot can speak |
| <code>Constructor()</code> | calls the parent constructor using its parameter values; initialises <code>WingSpan</code> and <code>NumberWords</code> to its parameter values |
| <code>ChangeNumberWords()</code> | takes an integer parameter and adds this to the number currently in <code>NumberWords</code> |

(i) Write program code to declare the class `Parrot`, its constructor and the method `ChangeNumberWords()`

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 **3(b)(i)** in the evidence document.

[4]

(ii) The method `Description()` in the class `Parrot` creates and returns a string of the animal's data in the format:

```
"The animal's name is " <Name> ", it makes a " <Sound> ", its size is " <Size> " and its intelligence level is " <Intelligence> ". It has a wingspan of " <WingSpan> "cm and can say " <NumberWords> " words."
```

For example:

```
The animal's name is Chewie, it makes a Squawk, its size is 1 and its intelligence level is 10. It has a wingspan of 30cm and can say 29 words.
```

Write program code for `Description()`

Save your program.

Copy and paste the program code into part **3(b)(ii)** in the evidence document.

[2]

(c) The class `Wolf` inherits from the class `Animal`

`Wolf` inherits the attributes from `Animal` and overrides the `Description()` method. The additional attributes and methods in the class `Wolf` are:

| Wolf | |
|--------------------------------------|--|
| <code>TerritorySize : Integer</code> | stores the size of the area where the wolf lives, to the nearest square mile |
| <code>Constructor()</code> | calls the parent constructor using its parameter values; initialises <code>TerritorySize</code> to its parameter value |
| <code>SetTerritorySize()</code> | takes an integer parameter and adds this to the number currently in <code>TerritorySize</code> |

(i) Write program code to declare the class `Wolf`, its constructor and the method `SetTerritorySize()`

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 **3(c)(i)** in the evidence document.

[4]

(ii) The method `Description()` in the class `Wolf` creates and returns a string of the animal's data in the format:

```
"The animal's name is " <Name> ", it makes a " <Sound> ", its size is " <Size> " and its intelligence level is " <Intelligence> ". Its territory is " <TerritorySize> " square miles."
```

For example:

```
The animal's name is Nighteyes, it makes a Howl, its size is 8 and its intelligence level is 7. Its territory is 100 square miles.
```

Write program code for `Description()`

Save your program.

Copy and paste the program code into part **3(c)(ii)** in the evidence document.

[2]

(d) (i) The main program declares instances of the classes for **three** animals:

- A parrot with the name 'Chewie'; it makes a 'Squawk' sound. Its size is 1, intelligence is 10, wingspan is 30 cm and it can say 29 words.
- A wolf with the name 'Nighteyes'; it makes a 'Howl' sound. Its size is 8, intelligence is 7 and its territory is 100 square miles.
- An animal that is a horse with the name 'Copper'; it makes a 'Neigh' sound. Its size is 10 and its intelligence is 6.

Write program code for the main program.

Save your program.

Copy and paste the program code into part **3(d)(i)** in the evidence document.

[2]

(ii) The main program also needs to:

- decrease the territory for the wolf Nighteyes by 20 square miles
- increase the number of words the parrot Chewie can say by 2 words
- output the description for all **three** animals.

Write program code to extend the main program.

Save your program.

Copy and paste the program code into part **3(d)(ii)** in the evidence document.

[3]

(iii) Test your program.

Take a screenshot of the output(s).

Save your program.

Copy and paste the screenshot into part **3(d)(iii)** in the evidence document.

[2]