

Past-paper walk-through

Programming Paradigms ■ 20.1

eXercise

Questions in this set

- 9618/33 J25 Q5 ■ 4 marks
- 9618/42 J25 Q3 ■ 26 marks
- 9618/43 J25 Q3 ■ 29 marks
- 9618/32 N24 Q10 ■ 6 marks
- 9618/33 N24 Q9 ■ 6 marks

Reading the mark scheme

- **B1** a mark that stands on its own – the point is either made or not.
- **M1** a *method* mark: the right approach, even if the number is wrong.
- **A1** an *answer* mark, and it usually needs its M mark first.
- **C1** a working mark on the way to the answer.
- **//** separates answers the examiner accepts equally.
- **ecf** = error carried forward: a later part is marked on your own earlier answer.
- **owtte** = “or words to that effect” – the wording need not match.

Question 5

9618/33 J25 ■ Q5 ■ 4 marks

Complete the table by filling in the missing object-oriented programming (OOP) terms and descriptions.

OOP term	Description
	A method that accesses the value of a property.
	A method that changes the value of a property.
Object	
Method	

Question 5

[4]

Question 5

OOP term	Description
.....	A method that accesses the value of a property.
.....	A method that changes the value of a property.
Object	
Method	

[4]

Solution 5

Worked steps

Each row is one mark, and the wording that scores is the one that says what the thing **is**, not what it is for.

OOP term	Purpose
Getter	A method that accesses (returns) the value of a property
Setter	A method that changes the value of a property
Object	An instance of a class
Inheritance	A class taking the properties and methods of another (parent) class

Solution 5

Two distinctions decide most of the marks here:

- A **class** is the template; an **object** is one instance built from it. Saying an object “is a class” scores nothing.
- Getters and setters exist because properties are **private**: encapsulation hides the data, and these methods are the controlled way in and out. Mentioning that is what turns a definition into an explanation if the question asks for one.

Solution 5

Mark scheme

- One mark per correct answer (Max 4)
- OOP Term
- Purpose
- Getter
- A method that accesses the value of a property
- Setter
- A method that changes the value of a property
- Object
- An instantiation / instance of a class
- Method

Solution 5

- A programmed function / procedure / subroutine
- / subprogram defined as part of a class

Question 3

9618/42 J25 ■ Q3 ■ 26 marks

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

The class `Animal` stores the data about animals.

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

Question 3

(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.

Question 3

[4]

(a)(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.

Question 3

[3]

Question 3

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

Question 3

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

Question 3

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>