# Assessment test Answers

# Section 1: Topics 1-2

1. (a) In object oriented programming, explain what is meant by:

(i) a class [1]

A blueprint for objects

(ii) a constructor [2]

A method/procedure which allows new objects in the class to be created

(iii) instantiation [1]

A statement which creates an object belonging to a particular class

(iv) polymorphism [2]

A programming language’s ability to process objects differently depending on their class (1) plus an example (1)

or

redefining a class method defined in a superclass, using the same method name but using different attributes and/or behaviours/instructions

(b) Explain **two** advantages of object-oriented programming over procedural programming. [4]

Encapsulation – classes can contain all the data and procedures needed. Programmers using the class cannot directly access variables and change them

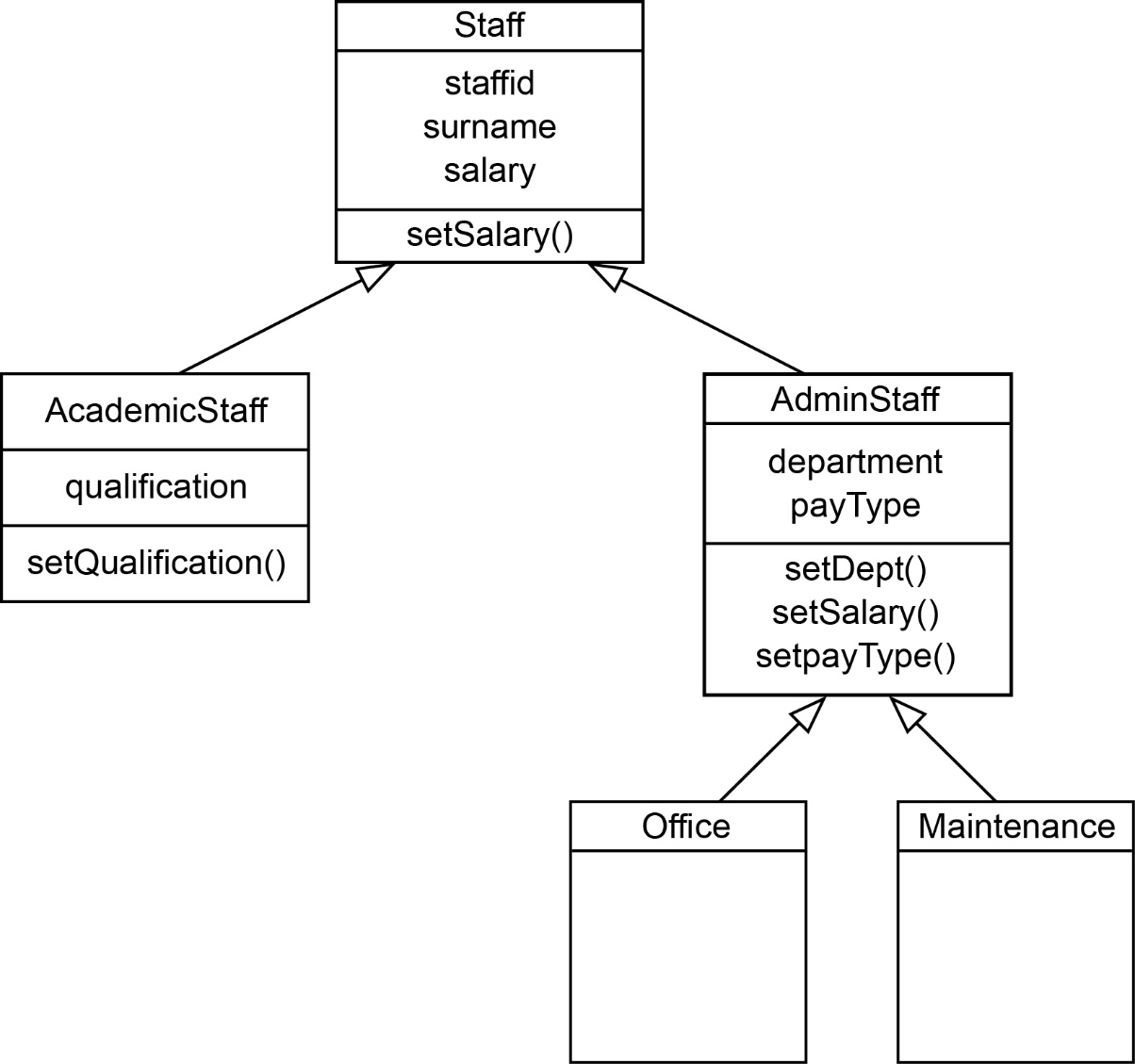
Code is reusable – class libraries can be imported into other programs

Decreases program complexity since much of the processing is encapsulated with its class

Maintenance becomes easier since there is less code to check when modules are reused

2. A school employs two different types of staff, Academic and Administrative.   
Administrative staff are categorised as Office or Maintenance staff.

An incomplete diagram of the system is shown below.



(a) What is this type of diagram called? [1]

class diagram

(b) State the terms that describe

(i) setSalary() Method [1]

(ii) staffed Attribute/property [1]

(c) Explain the meaning of the arrows in the diagram, using an example. [2]

They indicate that the classes AcademicStaff and AdminStaff and both subclasses of Staff, for example, and both inherit all the attributes and methods of Staff. /Staff is a superclass of AcademicStaff and AdminStaff. AdminStaff is in turn a superclass of Office and Maintenance.

These classes also have their own individual attributes and methods.

Office and Maintenance inherit the methods and attributes of AdminStaff.

(d) Assume that AC123 has been defined as an object belonging to   
class AcademicStaff.

Explain why the statement:

AC123.setSalary(30000)

is valid even though setSalary() is not shown in this part of the diagram. [2]

Any object in the class AcademicStaff can use any of the methods inherited   
from the superclass Staff, so it does not need to declare its own method.

(e) Academic staff are paid a monthly salary. Administrative staff are paid either   
weekly or monthly, depending on their payType.

Explain why setSalary() appears in the description of both Staff and AdminStaff. [3]

setSalary() is an example of polymorphism;

whereby a subclass can redefine the method defined in a superclass;

so that it will behave differently in the subclass even though called with the same name;

For example, the salary in AdminStaff depends on payType which may be monthly or weekly.

[Section 1 Total: 20 marks]

# Section 2: Topics 3-6 Answers

3. (a) If and , write an expression for   
(the composition of p and q). [1]

=

(b) Use functional composition to write code for a single function **comp x** which would implement p, q and .

You may assume that the built-in function **sqrt** finds the square root of a number. [3]

p x = sqrt (2 \* x)

q x = x + 5

comp x = p (q x) or comp x = (p.q) x

(c) What is the value of comp 3? [1]

4.0

(d) The **domain** of comp is the set of natural numbers ℕ, where ℕ = {0, 1, 2, 3…}.

What is the **co-domain** of comp? [1]

Real / Float / the set of real numbers ℝ

4. What is returned by the following operations on

**listT = ["Oak", "Ash", "Elm", "Yew", "Pine"]** [5]

(a) head (tail listT)

“Ash”

(b) tail (tail (head listT))

“k”

(c) map head listT

"OAEYP"

(d) length listT

5

(e) null (tail (tail (head listT)))

False (since tail (tail (head listT)) = “k”)

5. **Map**, **filter**, and **fold** functions are all higher order functions.

(a) Explain what is meant by a **higher order function**. [2]

A function is higher-order if it takes a function as an argument or returns a   
function as a result, or both.

(b) Explain how a **map** function can be used to apply a discount of 6% to a list of transaction amounts. [3]

A discount calculation can be expressed as a function (1 mark) such as

discount x = x \* 0.94 (1 mark)

which can then be applied to each element of a list (1 mark) with an expression such as

map (discount) listTransactions / accept map(\*0.94) listTransactions (1 mark)

*Max 3*

(c) Given listQ = [56, 14, 89, 23, 12, 7], how could you use one of the functions **map**, **filter**, **fold** to return all the elements in listQ which are less than 50? [2]

filter (<50) listQ

(d) Which function would you use to find the sum of all the elements in listQ? [1]

fold

*In Haskell, foldl (+) 0 listQ*

6. In a certain functional language, **listA** is defined using the statement

A = [1, 7, 6]

(a) What will be the output after each of the following consecutive operations?

(i) append [4, 5] to listA [1]

[1, 7, 6, 4, 5]

(ii) prepend [9] to listA [1]

[9, 1, 7, 6]

*(Note that the original list remains unchanged)*

(iii) length[listA] [1]

3

(b) Write a statement to replace the first two numbers of listA with 17 and 11 [3]

prepend [17,11] to tail (tail listA ))

7. Big Data can be described in terms of **volume**, **velocity** and **variety**. Give an   
example of a big Data application and use it to describe what is meant by these three terms. [7]

Any example of big data such as Google, social media site, smartphone data, retailer such as Amazon, data from sensors, with description of data that needs to be processed. max 3

Volume – too big to fit on a single server. max 1

Velocity – response needs to be given within a few milliseconds, description of type of response in given example max 2

Variety – data is unstructured and may be in different forms such as text, images, multimedia / not suitable for traditional row and column databases. max 2

8. Describe features of functional programming which makes it particularly suitable   
for handling Big Data applications. [3]

Data / data structures in a functional language are immutable / there are no side-effects of computations (1)

one part of a functional program cannot change data and thus affect another part (1)

Functional operations such as map and fold / reduce (1)

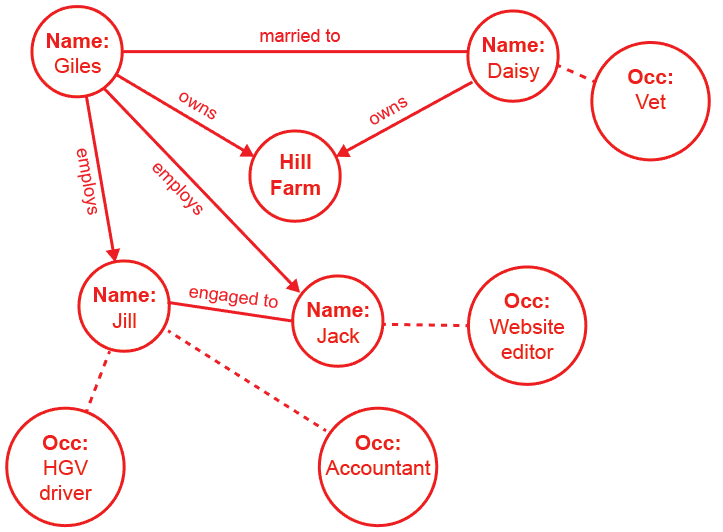
can be easily parallelised (1) for efficient (1) execution

Order of execution less rigidly defined in a functional language than for procedural/object-oriented/other paradigms (1)

It is easier to prove correctness of a functional programme (1) *max 4*

9. Farmer Giles owns Hill Farm jointly with his wife Daisy, a vet. Farmer Giles employs Jack the cowman, who also edits the farm website, and Jill who keeps the accounts and is engaged to Jack. Jill is an HGV driver.

Sketch a **graph schema** to represent these facts. [5]



**Occ:**

Cowman

Marks: 1 each for

* Married & engaged as undirected edges
* Owns as directed edges
* Employs as 2 directed edges
* Jill: 2 additional facts
* Daisy and Jack: additional facts

[Section 2 Total: 40 marks]

[Total for Sections 1 and 2: 60 marks]