# Worksheet 2 Object-oriented design principles

**Task 1**

Program the superclass Animal and the subclasses Fish and Duck and test your code.

Pseudocode

Animal = Class

 Protected

 state: String

 size: Integer

 Public

 Constructor(s, n)

 state 🡨 s

 size 🡨 n

 End Constructor

 Procedure feed()

 size 🡨 size + 1

 Output state, " fed"

 End Procedure

 Function getState()

 Return state

 End Function

 Function getSize()

 Return size

 End Function

End Class

Fish = SubClass(Animal)

 Private

 maxSize : Integer

 Public

 Constructor(s)

 Animal.Constructor(s, 1)

 maxSize = 2

 End Constructor

 Procedure setMaxSize(m)

 maxSize 🡨 m

 End Procedure

 Procedure feed Override

 size += 2

 Output state, " fed"

 If size >= maxSize Then

 state 🡨 "BIG FISH"

 End If

 End Procedure

End Class

Duck = Subclass(Animal)

 Public

 Procedure feed Override

 Animal.feed()

 If size = 5 Then

 state 🡨 "BIG DUCK"

 End If

 End Procedure

End Class

thisFish 🡨 new Fish("little fish", 1)

thisFish.setMaxSize(3)

thisDuck 🡨 new Duck("little duck", 1)

For count 🡨 1 To 3

 thisDuck.feed()

 Output(thisDuck.getState())

 thisFish.feed()

 Output(thisFish.getState())

End For

**Task 2**

Draw a class diagram for a superclass Vehicle and subclasses Car and Lorry.

All classes have attributes VehicleID, Mileage (number of whole miles), TankContents (number of litres).

The Car class also has attributes NumberOfSeats.

The Lorry class also has the attributes MaxLoadWeight (in kg) and CurrentLoad (in kg).

Methods for the superclass are Drive, Refuel and CalculateUsage.

The Car class has the additional method AddPassengers.

The Lorry class has additional methods of AddLoad and Unload.

The subclasses implement the CalculateUsage method differently and require access to the Mileage attribute.

**Task 3**

Draw a diagram of a ComputerSystem class that is composed of the following classes:

Display, Keyboard, Memory, CPU, HardDrive, Printer, Motherboard.

**Task 4**

Program the classes Order, Order Status and Product and test your code.

Pseudocode

Order = Class

 Private

 OrderNumber: String

 OrderDate: Date

 ProductsOrdered: Array Of Product

 NumberOfItemsOrdered: Integer

 Status: OrderStatus

End Class

OrderStatus = Class

 Private

 HasShipped: Boolean

End Class

Product = Class

 Private

 ProductID: String

 ProductPrice: Currency

End Class

product1 🡨 new Product("beans", 0.45)

product2 🡨 new Product("eggs", 1.25)

myOrder 🡨 new Order(1, "1/1/17")

myOrder.OrderItem(product1)

myOrder.OrderItem(product2)

Output(myOrder.getOrderStatus())

Output(myOrder.getOrderItemID(1))

Output(myOrder.getOrderItemPrice(1))

Output(myOrder.getOrderItemID(2))

Output(myOrder.getOrderItemPrice(2))

**Note: you will need to add the required constructor and method definitions**