Past Papers May/June 2015 to 2018:

9608/41/M/J/15

4 A payroll program is to be written using an object-oriented programming language. An

Employee class is designed. Two subclasses have been identified:

- `HourlyPaidEmployee` who is paid a monthly wage calculated from their hourly rate of pay and the number of hours worked during the month.
- `SalariedEmployee` who is paid a monthly wage which is one 12th of their annual salary (a) Draw an inheritance diagram for these classes.

(a) Draw an inheritance diagram for these classes.

[3]

(b) The design for the Employee class consists of:

- **properties**
  - `EmployeeName`
  - `EmployeeID`
  - `AmountPaid`
  - `ThisMonth`
- **methods**
  - `SetEmployeeName`
  - `SetEmployeeID`
  - `CalculatePay`

Write program code for the class definition of the superclass Employee.

Programming language ....

....

....

....

....

....

....

....

....

....

.... [5]

(c) (i) State the properties and/or methods required for the subclass

HourlyPaidEmployee.

....

....

....

.... [4]

(ii) State the properties and/or methods required for the subclass SalariedEmployee.

....

....

....

.... [2]

(d) Name the feature of object-oriented program design that allows the method

`CalculatePay` to be declared in the superclass `Employee`.

....

.... [1]

9608/43/M/J/15

Q4 A sports club stores data about its members. A program is to be written using an object-oriented programming language.

A **`Member`** class is designed. Two subclasses have been identified:

- **`FullMember`**
- **`JuniorMember`**

(a) Draw an inheritance diagram for these classes.

[3]

(b) The design for the Member class consists of

- **Properties**
  - `MemberName`
  - `MemberID`
  - `SubscriptionPaid`
- **Methods**
  - `SetMemberName`
  - `SetMemberID`
  - `SetSubscriptionPaid`

Write program code for the class definition of the superclass `Member`.

Programming language ....

....
....
....
....
....
....
....
....
....
....[5]

(c) Additionally a **`DateOfBirth`** property is required for the **`JuniorMember`** class.

(i) Write program code for the class definition for the subclass **`JuniorMember`**.

....
....
....
....
....
.... [3]

(ii) Write program code to create a new instance of **`JuniorMember`**.

Use identifier **`NewMember`** with the following data:

name Ahmed with member ID 12347, born on 12/11/2001, who has paid his subscription.

....
....
....
....
....
.... [3]

9608/42/M/J/18

Q4 X-Games is an international extreme sports competition. A program will store and process data about the teams in the competition.

- Each team is made up of members.
- Members can be added and removed from each team.
- Each member has a first name, last name, date of birth and gender.
- Each member can be an official or a competitor.
- Each official has a job title and may be first-aid trained.
- Each competitor takes part in one sport.

The program is written using object-oriented programming. The program can output the full name and date of birth of any member. For example, “Nadia Abad 16/05/1995”

An introduction about a team member can be output using their name. For example, “Hello, I’m Nadia Abad”.

The program outputs a different version of the introduction for a competitor. This version includes the competitor’s sport. For example, “Hello, I’m Sally Jones and my sport is Skateboard Park.”

(a) Complete the following class diagram to show the attributes, methods and inheritance for the program.
You do not need to write the get and set methods.

Member
FirstName : STRING LastName : STRING DateOfBirth : DATE Gender : STRING

Constructor()
Introduction()
DisplayFullnameAndDateOfBirth()

Team
TeamName : STRING TeamList : ARRAY OF Member
Constructor() ....

Competitor
Sport : STRING

Constructor()
Introduction()

Official
....

Constructor()
DisplayJobTitle()

[3]

(b) Write program code for the Member class.
Programming language ....
Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(c) Write program code for the Competitor class.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[5]

(d) Omar Ellaboudy is an official at X-Games. He is first-aid trained and his job title is Judge. He is male and was born on 17/03/1993.

Write program code to create an instance of an object with the identifier BMXJudge.

All attributes of the instance must be fully initialised.

Programming language ....

Program code

....

....

....

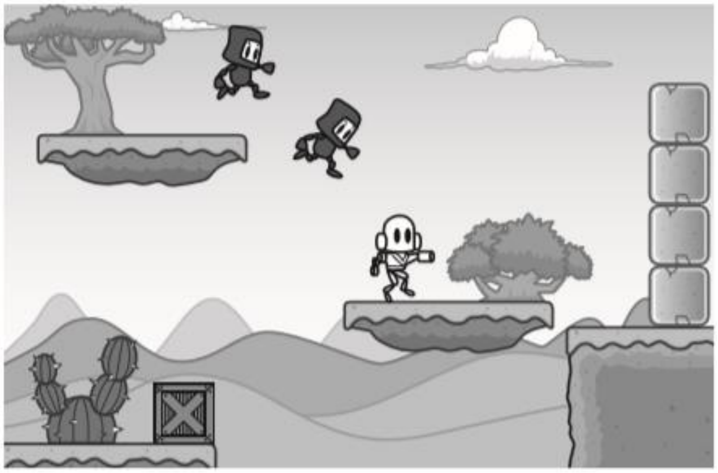
....

....

....[3]

9608/43/M/J/18

5 A computer game is being developed using object-oriented programming. The following image is a screenshot from the game.



There are scenery elements and animated elements. The player's character is one of the animated elements.

Each game element has the attributes:

Attribute	Description	Example value
`PositionX`	The x coordinate of the game element.	`92`
`PositionY`	The y coordinate of the game element.	`106`
`Width`	The width of the game element.	`150`
`Height`	The height of the game element.	`200`
`ImageFilename`	The filename of the image file for the game element.	`GameElementFrame1.png`

Each game element has a method, `GetDetails()` that returns a string containing all the element's attributes.

The player's character is one of a number of animated elements. All animated elements have the attributes:

Attribute	Description	Example value
AnimationFrames	An array of GameElement	
Direction	A string giving the direction the object is travelling in.	"Left"
Strength	A value for the strength that indicates the power of the object.	2000
Health	A value for the health that indicates the health of the object.	100

The player's character can either move left or right, or jump.

(a) Complete the following class diagram for the game. You do not need to include any additional get or set methods.

```
GameElement
-----------------------------
PositionX: INTEGER
PositionY: INTEGER
Width: INTEGER
Height: INTEGER
ImageFilename: STRING
-----------------------------
Constructor()
GetDetails()
```

```
AnimatedElement
-----------------------------
AnimationFrames: ARRAY OF GameElement
.....................................
.....................................
.....................................
-----------------------------
Constructor()
AdjustHealth()
AdjustStrength()
DisplayAnimation()
```

```
Scenery
-----------------------------
CauseDamage: BOOLEAN
DamagePoints: INTEGER
-----------------------------
Constructor()
GiveDamagePoints()
```

```
Player
-----------------------------

-----------------------------
.....................................
.....................................
.....................................
.....................................
```

[3]

(b) Write program code to define the GameElement class.

Programming language ....

Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[6]

(c) The `Scenery()` class has two attributes, `CauseDamage` and `DamagePoints`.

If the attribute `CauseDamage` is `TRUE`, then the scenery element can cause damage.

The method `GiveDamagePoints()` checks whether the object can cause damage.

If the object can cause damage, the method returns the integer value of the `DamagePoints` attribute.
Write program code for the `Scenery class`.
Programming language
....
Program code

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

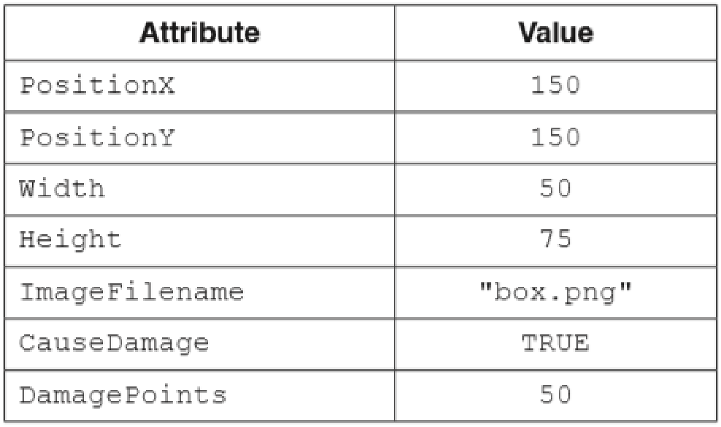
....

....

....

....[6]

(d) A new scenery object, `GiftBox`, is to be created.
(i) The attributes of `GiftBox` are as follows:



Attribute	Value
`PositionX`	`150`
`PositionY`	`150`
`Width`	`50`
`Height`	`75`
`ImageFilename`	`"box.png"`
`CauseDamage`	`TRUE`
`DamagePoints`	`50`

Write program code to create an instance of `GiftBox`.

Programming language

....

Program code

....

....

....

....

....

....

....[3]

(ii) An additional method, `GetScenery()`, returns all the attributes of the Scenery class.

Write program code for the `GetScenery()` method. You should use the `GetDetails()` method that the Scenery class inherits from the `GameElement` class.

Programming language

....

Program code

....

....

....

....

....

....

....

....

....

....

....

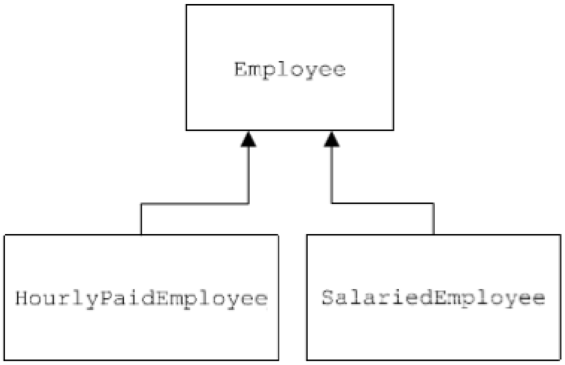
....

....

....[3]

Answers:
9608/41/M/J/15

4 (a)



[3]

(b) Example VB

Mark as follows:

Class header	(1 mark)
PUBLIC and PRIVATE used correctly	(1 mark)
EmployeeName + EmployeeID	(1 mark)
AmountPaidThisMonth	(1 mark)
Methods x 3	(1 mark)

Example VB

```
Class Employee
    Private EmployeeName As String
    Private EmployeeID As String
    Private AmountPaidThisMonth As Decimal
Public Sub SetEmployeeName()
End Sub
Public Sub SetEmployeeID()
End Sub
Public Sub CalculatePay()
End Sub
```

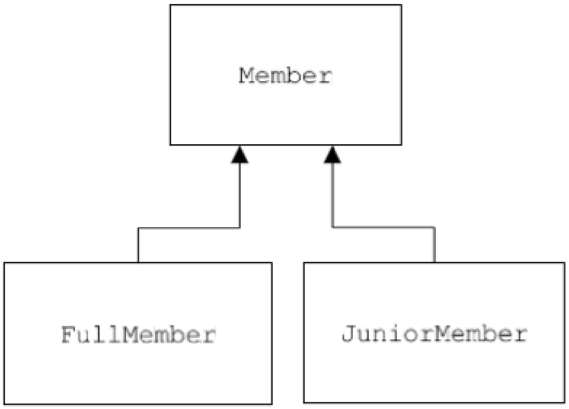
[max 5]

(c)	**(i)**	HoursWorked	1	
		HourlyPayRate	1	
		SetHoursWorked	1	
		CalculatePay : Override	1 + 1	
		SetPayRate	1	**[max 4]**
	(ii)	AnnualSalary	1	
		SetSalary	1	
		CalculatePay : Override	1	**[max 2]**

(d) Polymorphism **[1]**

Answers:
9608/43/M/J/15

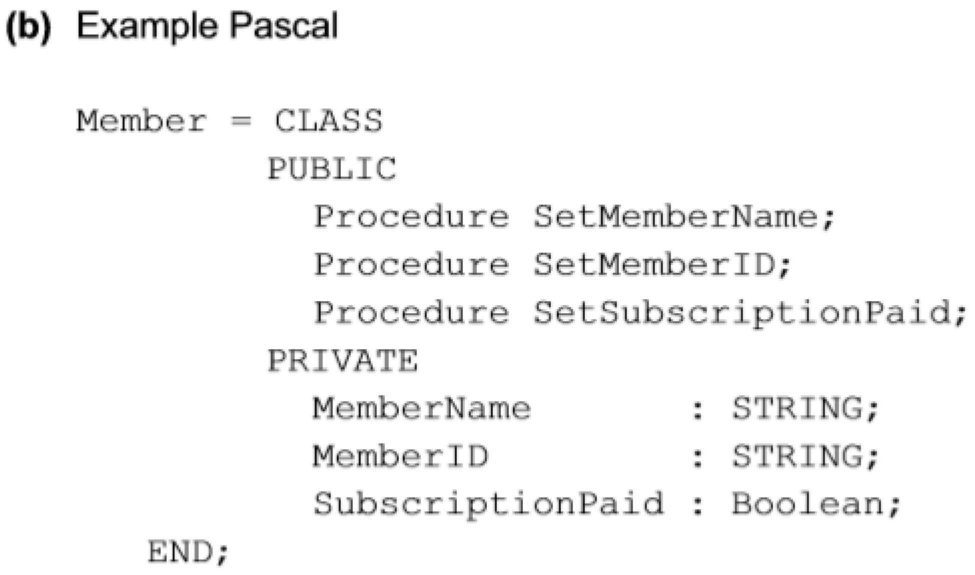
4 (a)



[3]

(b) Example VB

```
Class Member
     Private MemberName As String
     Private MemberID As String
     Private SubscriptionPaid As Boolean
Public Sub SetMemberName( )
End Sub
Public Sub SetMemberID ( )
End Sub
Public Sub SetSubscriptionPaid ( )
End Sub
```



(b) Example Pascal

```
    Member = CLASS
               PUBLIC
                 Procedure SetMemberName;
                 Procedure SetMemberID;
                 Procedure SetSubscriptionPaid;
               PRIVATE
                 MemberName       : STRING;
                 MemberID         : STRING;
                 SubscriptionPaid : Boolean;
         END;
```

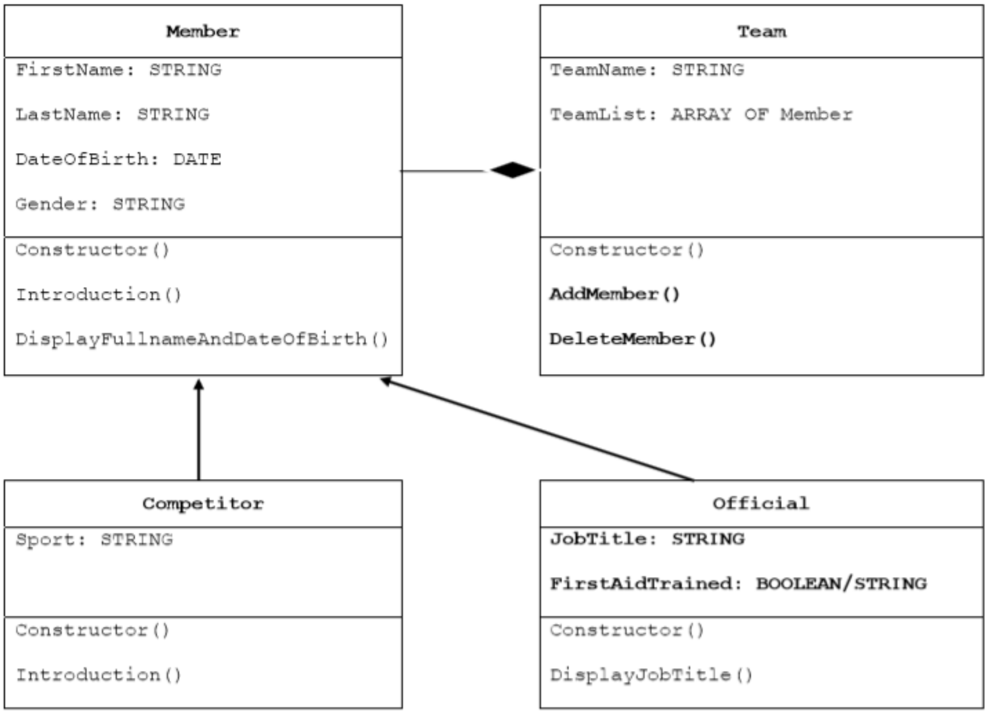
Mark as follows:

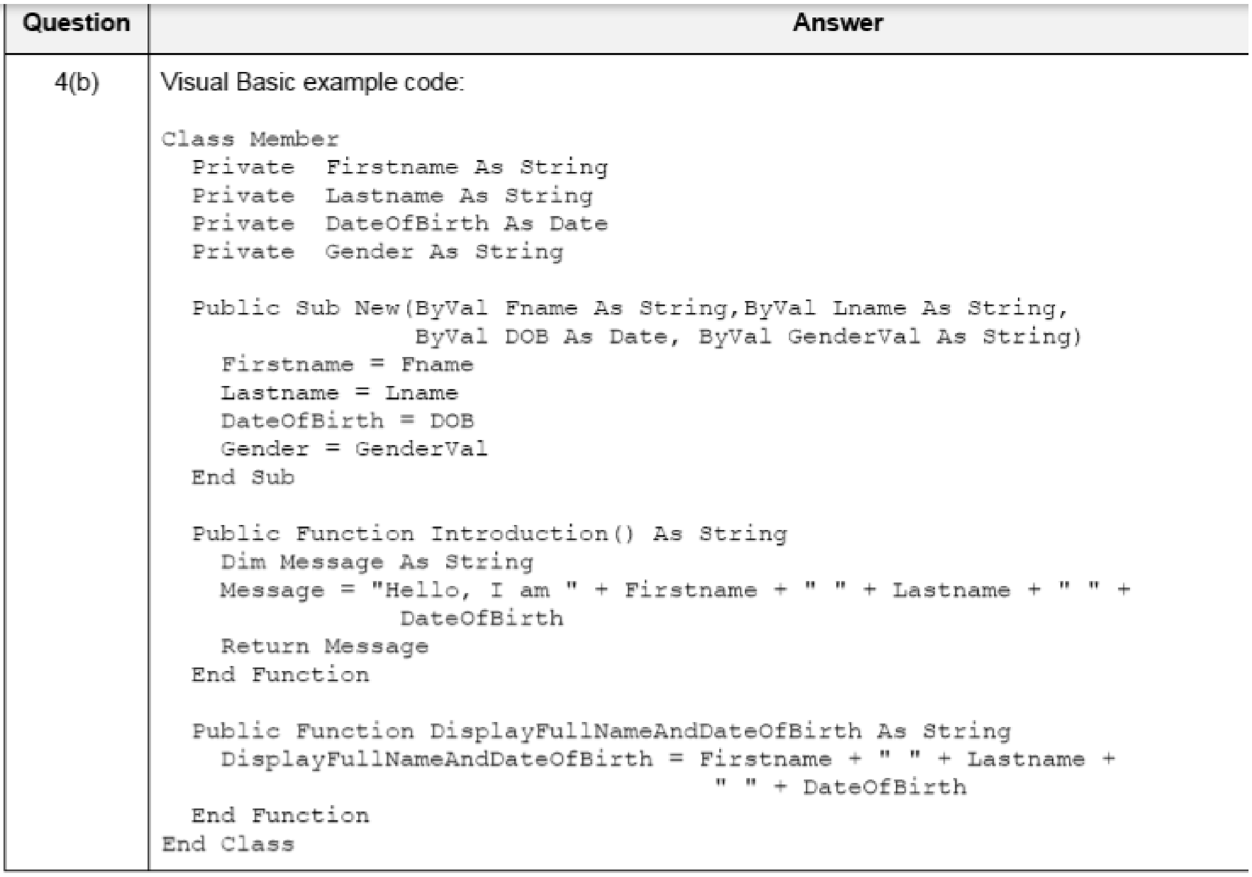
Class header	(1 mark)	
`Public` and `Private` used correctly	(1 mark)	
`MemberName` + `MemberID`	(1 mark)	
`SubscriptionPaid`	(1 mark)	
Methods × 3	(1 mark)	**[5]**

(c) (i) Example Pascal

```
JuniorMember = CLASS (Member)
                    PUBLIC
                       Procedure SetDateOfBirth;
                    PRIVATE
                       DateOfBirth : DateTime;
                 END;
```

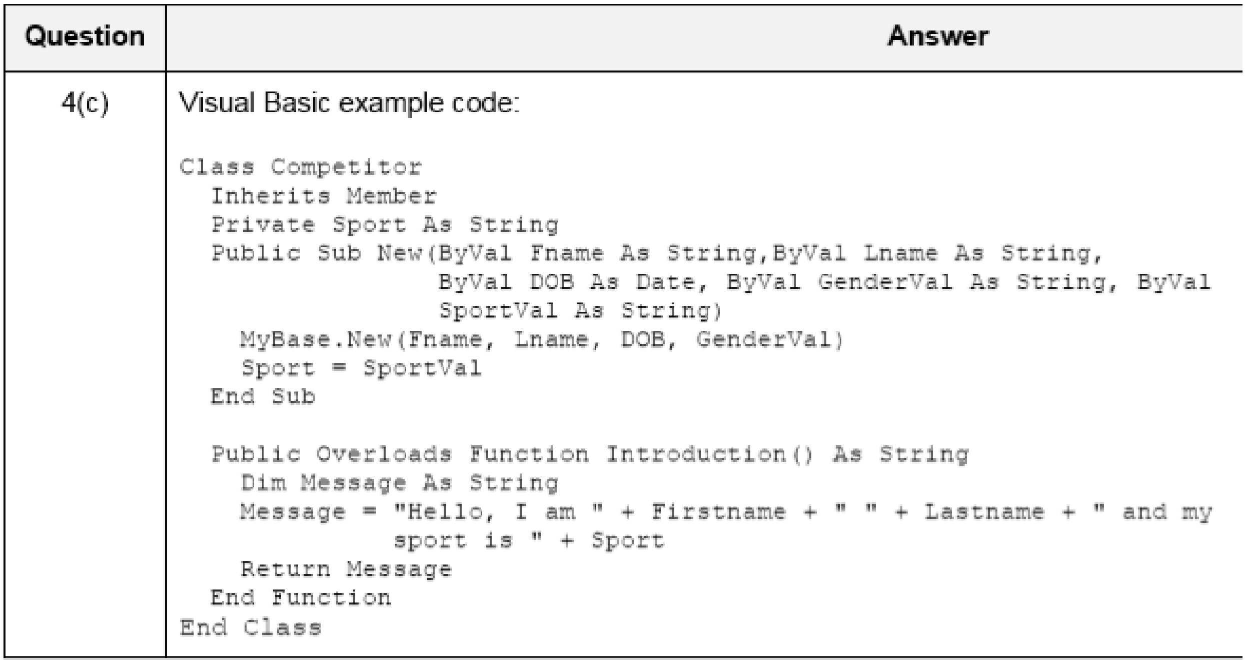
Answers:
9608/42/M/J/18

Question	Answer
4(a)	1 mark per bullet: • Team methods • Official attributes • Two inheritance arrows **or** containment  Member FirstName: STRING LastName: STRING DateOfBirth: DATE Gender: STRING Constructor() Introduction() DisplayFullnameAndDateOfBirth() Team TeamName: STRING TeamList: ARRAY OF Member Constructor() AddMember() DeleteMember() Competitor Sport: STRING Constructor() Introduction() Official JobTitle: STRING FirstAidTrained: BOOLEAN/STRING Constructor() DisplayJobTitle()



Question	Answer
4(b)	Visual Basic example code:

```
Class Member
  Private  Firstname As String
  Private  Lastname As String
  Private  DateOfBirth As Date
  Private  Gender As String

  Public Sub New(ByVal Fname As String,ByVal Lname As String,
               ByVal DOB As Date, ByVal GenderVal As String)
    Firstname = Fname
    Lastname = Lname
    DateOfBirth = DOB
    Gender = GenderVal
  End Sub

  Public Function Introduction() As String
    Dim Message As String
    Message = "Hello, I am " + Firstname + " " + Lastname + " " +
              DateOfBirth
    Return Message
  End Function

  Public Function DisplayFullNameAndDateOfBirth As String
    DisplayFullNameAndDateOfBirth = Firstname + " " + Lastname +
                                    " " + DateOfBirth
  End Function
End Class
```



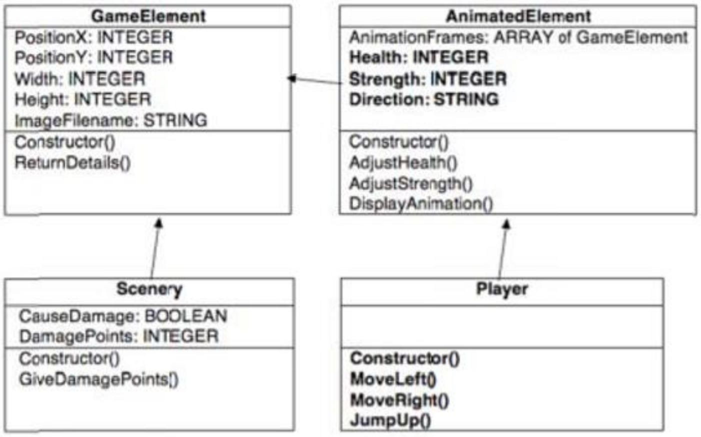
Question	Answer
4(c)	Visual Basic example code:

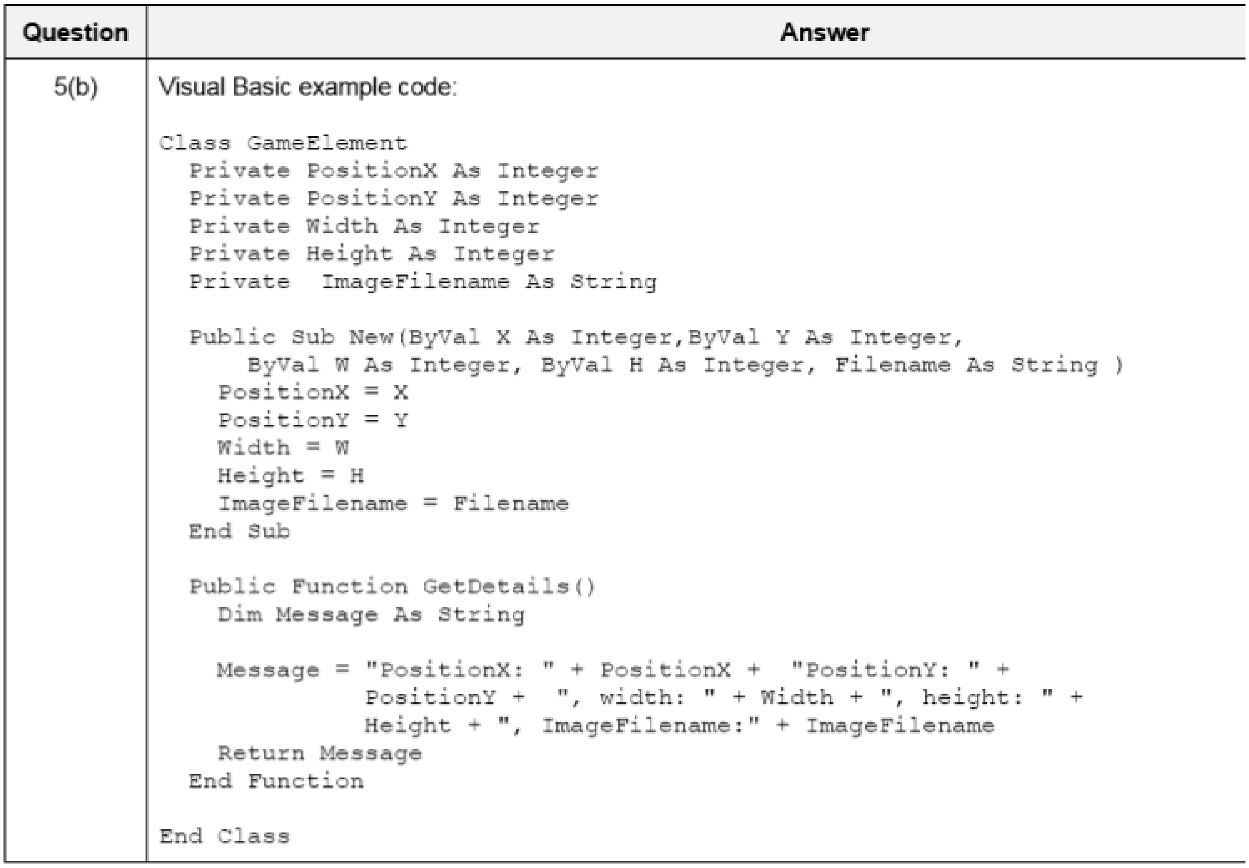
```
Class Competitor
  Inherits Member
  Private Sport As String
  Public Sub New(ByVal Fname As String,ByVal Lname As String,
               ByVal DOB As Date, ByVal GenderVal As String, ByVal
               SportVal As String)
    MyBase.New(Fname, Lname, DOB, GenderVal)
    Sport = SportVal
  End Sub

  Public Overloads Function Introduction() As String
    Dim Message As String
    Message = "Hello, I am " + Firstname + " " + Lastname + " and my
              sport is " + Sport
    Return Message
  End Function
End Class
```

Question	Answer	Marks

4(d)	1 mark per bullet
	• variable BMXJudge assigned value
	• call Official
	• with all 6 parameters assigned correctly
	Python example code:
	`BMXJudge = Official("Omar", "Ellaboudy", "17/03/1993", "Male", true, "Judge")`
	Visual Basic example code:
	`BMXJudge = New Official("Omar", "Ellaboudy", "17/03/1993", "Male, true, "Judge")`
	Pascal example code:
	`BMXJudge := Official("Omar", "Ellaboudy", "17/03/1993", "Male", true, "Judge")`
		3

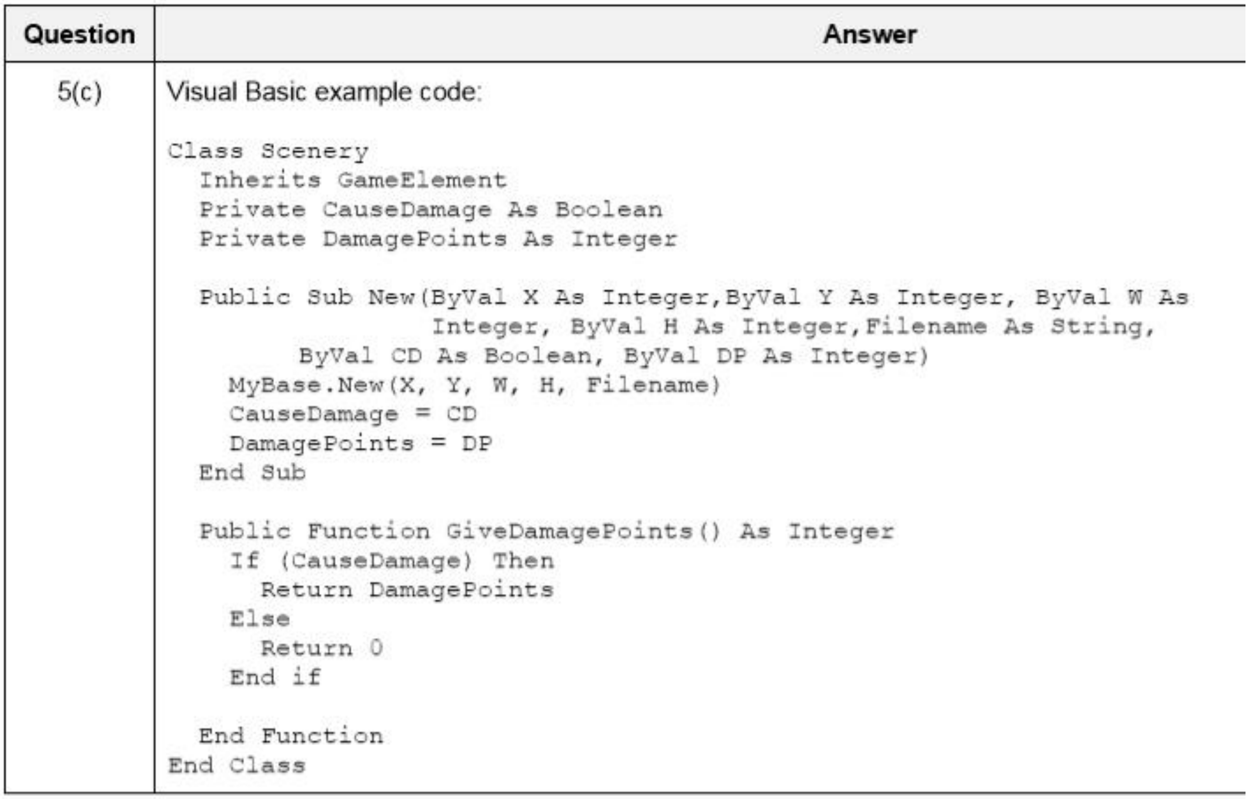
Answers:
9608/43/M/J/18

Question	Answer	Marks
5(a)	1 mark for each bullet: • AnimatedElement attributes • Player methods • Inheritance arrows  GameElement PositionX: INTEGER PositionY: INTEGER Width: INTEGER Height: INTEGER ImageFilename: STRING Constructor() ReturnDetails() AnimatedElement AnimationFrames: ARRAY of GameElement Health: INTEGER Strength: INTEGER Direction: STRING Constructor() AdjustHealth() AdjustStrength() DisplayAnimation() Scenery CauseDamage: BOOLEAN DamagePoints: INTEGER Constructor() GiveDamagePoints() Player Constructor() MoveLeft() MoveRight() JumpUp()	3



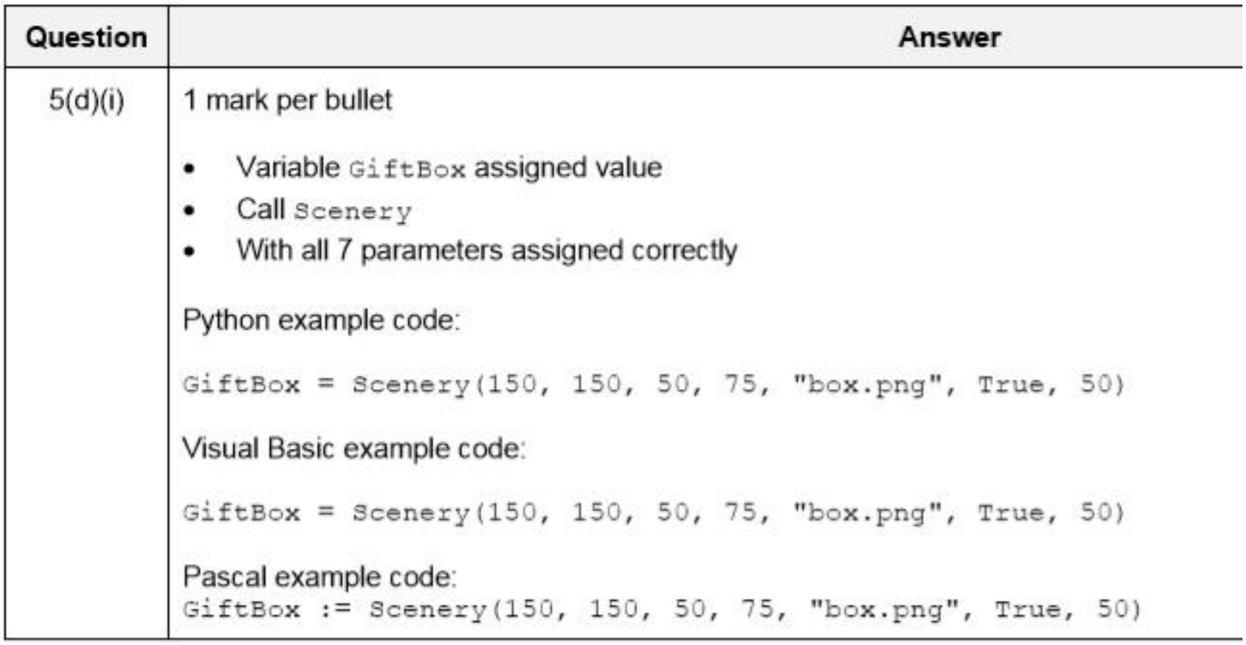
Question	Answer
5(b)	Visual Basic example code:

```
Class GameElement
  Private PositionX As Integer
  Private PositionY As Integer
  Private Width As Integer
  Private Height As Integer
  Private  ImageFilename As String

  Public Sub New(ByVal X As Integer,ByVal Y As Integer,
      ByVal W As Integer, ByVal H As Integer, Filename As String )
    PositionX = X
    PositionY = Y
    Width = W
    Height = H
    ImageFilename = Filename
  End Sub

  Public Function GetDetails()
    Dim Message As String

    Message = "PositionX: " + PositionX +  "PositionY: " +
              PositionY +  ", width: " + Width + ", height: " +
              Height + ", ImageFilename:" + ImageFilename
    Return Message
  End Function

End Class
```



Question	Answer
5(c)	Visual Basic example code:

```
Class Scenery
  Inherits GameElement
  Private CauseDamage As Boolean
  Private DamagePoints As Integer

  Public Sub New(ByVal X As Integer,ByVal Y As Integer, ByVal W As
                Integer, ByVal H As Integer,Filename As String,
         ByVal CD As Boolean, ByVal DP As Integer)
    MyBase.New(X, Y, W, H, Filename)
    CauseDamage = CD
    DamagePoints = DP
  End Sub

  Public Function GiveDamagePoints() As Integer
    If (CauseDamage) Then
      Return DamagePoints
    Else
      Return 0
    End if

  End Function
End Class
```



Question	Answer
5(d)(i)	1 mark per bullet • Variable `GiftBox` assigned value • Call `Scenery` • With all 7 parameters assigned correctly

Python example code:

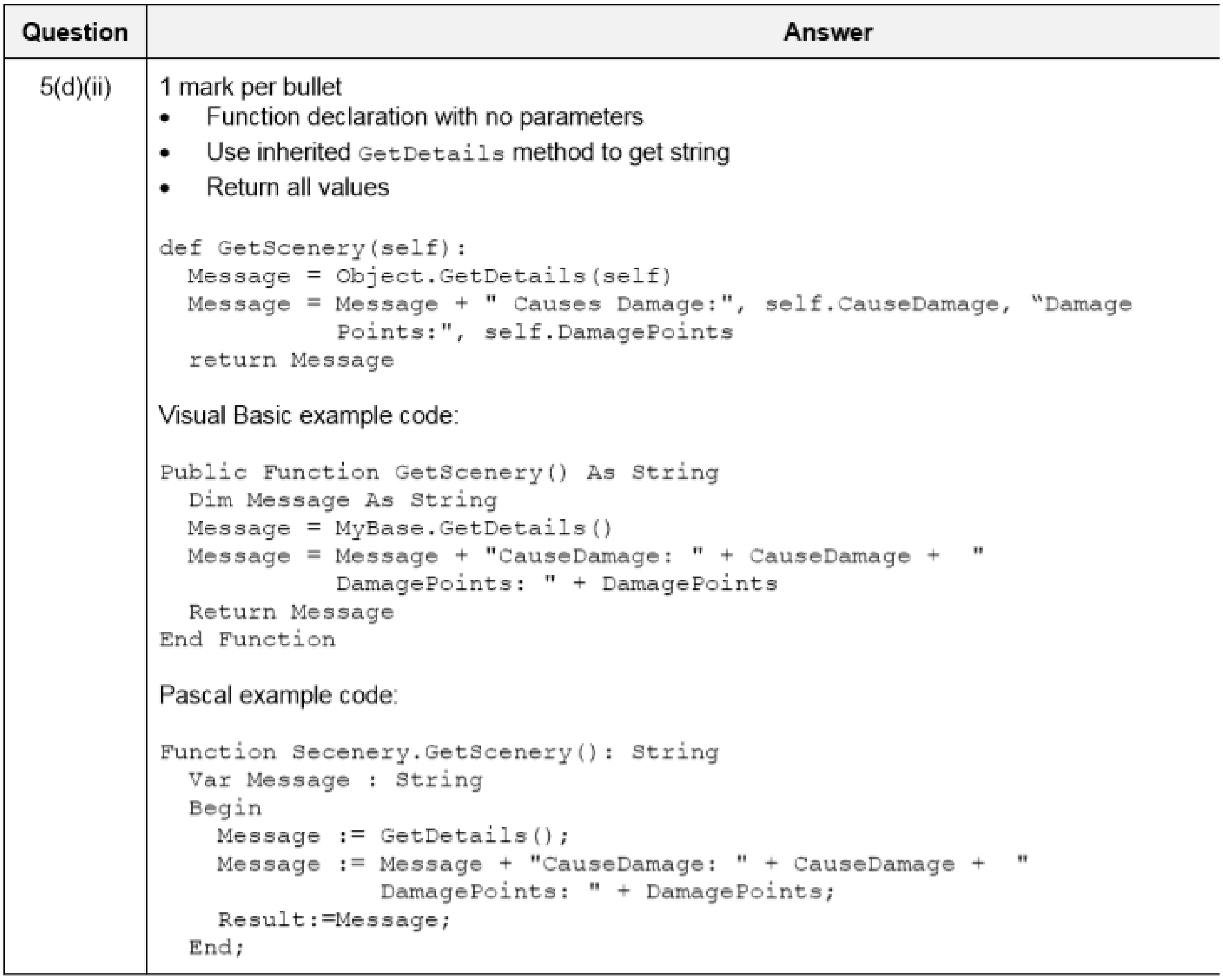
```
GiftBox = Scenery(150, 150, 50, 75, "box.png", True, 50)
```

Visual Basic example code:

```
GiftBox = Scenery(150, 150, 50, 75, "box.png", True, 50)
```

Pascal example code:

```
GiftBox := Scenery(150, 150, 50, 75, "box.png", True, 50)
```



Question	Answer
5(d)(ii)	1 mark per bullet • Function declaration with no parameters • Use inherited `GetDetails` method to get string • Return all values

```
def GetScenery(self):
  Message = Object.GetDetails(self)
  Message = Message + " Causes Damage:", self.CauseDamage, "Damage
            Points:", self.DamagePoints
  return Message
```

Visual Basic example code:

```
Public Function GetScenery() As String
  Dim Message As String
  Message = MyBase.GetDetails()
  Message = Message + "CauseDamage: " + CauseDamage +  "
            DamagePoints: " + DamagePoints
  Return Message
End Function
```

Pascal example code:

```
Function Secenery.GetScenery(): String
  Var Message : String
  Begin
    Message := GetDetails();
    Message := Message + "CauseDamage: " + CauseDamage +  "
               DamagePoints: " + DamagePoints;
    Result:=Message;
  End;
```