

QUESTION 1.



5 (a)

MembershipFile

Address	MemberID	other member data
0	0	
1	1001	
2	7002	
3	0	
4	0	
5	3005	
6	0	
7	0	
8	0	
:	:	
:	:	
96	4096	
97	0	
98	2098	
99	0	

1001 and 7002 and 3005	1
4096 and 2098	1



(b) (i) 10 // generate record address
 20 NewAddress ← Hash(NewMember.MemberID)
 30 // move pointer to the disk address for the record
 40 SEEK NewAddress
 50 PUTRECORD "MembershipFile", NewMember

[4]

(ii) 01 TRY
 02 OPENFILE "MembershipFile" FOR RANDOM
 03 EXCEPT
 04 OUTPUT "File does not exist"
 05 ENDTRY

[2]

(iii) collisions/synonyms
 The previous record will be overwritten

[2]

(iv) Create an overflow area
 The 'home' record has a pointer to others with the same key
OR
 Store the overflow record at the next available address
 in sequence
OR
 Re-design the hash function
 to generate a wider range of indexes // to create fewer collisions

[2]

(v) 41 GETRECORD "MembershipFile", CurrentRecord
 42 WHILE CurrentRecord.MemberID <> 0
 43 NewAddress ← NewAddress + 1
 44 IF NewAddress > 99 THEN NewAddress ← 0
 45 SEEK NewAddress
 46 GETRECORD "MembershipFile", CurrentRecord
 47 ENDWHILE

[max. 4]

QUESTION 2.



- 6 (a) 1 mark for structure header/ending
1 mark for each field correct, take away 1 mark for additional fields
Python answers will use a class

Pascal

```
TYPE StockItem = RECORD
    ProductCode      : String;    // accept integer
    Price            : Currency;  // accept real
    NumberInStock    : Integer;
END;
```

Python

```
class StockItem :
    def __init__(self) :
        self.ProductCode = ""      # = 0
        self.Price = 0.0           # = 0
        self.NumberInStock = 0
```

VB.NET

```
STRUCTURE StockItem
    Dim ProductCode As String    \ accept integer
    Dim Price As Decimal        \ Double/single
    Dim NumberInStock As Integer
END STRUCTURE
```

VB6

```
Type StockItem
    ProductCode As String    \ accept integer
    Price As Currency        \ Double/single
    NumberInStock As Integer
END Type
```

Page 12	Mark Scheme	
	Cambridge International A Level – October/November 2015	



(b) (i) 01 TRY
02 OPENFILE "StockFile" FOR READ/RANDOM // ignore "
03 EXCEPT
04 OUTPUT "File does not exist"
05 ENDTRY

- (ii) (Line 01) alerts system to check for possible run-time errors (exception)
(Lines 03, 04) handle the exception without the program crashing // keeps program running// provide alternative statements to execute to avoid run-time error

Accept "exception handling" for 1 mark

[Max 2]

(c) WHILE NOT EOF("StockFile")
 READFILE "StockFile", ThisStockItem // accept reading separate fields
 OUTPUT ThisStockItem.ProductCode
 OUTPUT ThisStockItem.NumberInStock
ENDWHILE

1 mark for loop (accept REPEAT)

1 mark for EOF("StockFile") // StockFile.Peek <> -1 / NONE/"

1 mark for READ record

1 mark for OUTPUT of 2 fields

Ignore opening and closing file

[4]

QUESTION 3.



5 (a)

MembershipFile

Address	MemberID	other member data
0	0	
1	1001	
2	7002	
3	0	
4	0	
5	3005	
6	0	
7	0	
8	0	
:	:	
:	:	
96	4096	
97	0	
98	2098	
99	0	

1001 and 7002 and 3005

4096 and 2098

1

1



(b) (i) 10 // generate record address
 20 NewAddress ← Hash(NewMember.MemberID)
 30 // move pointer to the disk address for the record
 40 SEEK NewAddress
 50 PUTRECORD "MembershipFile", NewMember

[4]

(ii) 01 TRY
 02 OPENFILE "MembershipFile" FOR RANDOM
 03 EXCEPT
 04 OUTPUT "File does not exist"
 05 ENDTRY

[2]

(iii) collisions/synonyms
 The previous record will be overwritten

[2]

(iv) Create an overflow area
 The 'home' record has a pointer to others with the same key
OR
 Store the overflow record at the next available address
 in sequence
OR
 Re-design the hash function
 to generate a wider range of indexes // to create fewer collisions

[2]

(v) 41 GETRECORD "MembershipFile", CurrentRecord
 42 WHILE CurrentRecord.MemberID <> 0
 43 NewAddress ← NewAddress + 1
 44 IF NewAddress > 99 THEN NewAddress ← 0
 45 SEEK NewAddress
 46 GETRECORD "MembershipFile", CurrentRecord
 47 ENDWHILE

[max. 4]

Question	Answer	Marks
3(a)	1 mark per bullet • Instantiation of island object and calling DisplayGrid • Loop 3 times and Island.HideTreasure • Call procedures StartDig and DisplayGrid Example Python <pre> Island = IslandClass() DisplayGrid() for Treasure in range(3): Island.HideTreasure() StartDig() DisplayGrid() </pre> Example Pascal <pre> var Island : IslandClass; var Treasure : integer; begin Island := IslandClass.Create(); DisplayGrid; for Treasure := 1 to 3 do Island.HideTreasure(); StartDig; DisplayGrid; end; </pre>	3 1 1 1 1 1 1

Question	Answer	Marks
	Example VB.NET <pre> Dim Island As New IslandClass() DisplayGrid() For Treasure = 1 To 3 Island.HideTreasure() Next StartDig() DisplayGrid() </pre>	1 1 1

PUBLISHED

Question	Answer	Marks
3(b)	1 mark per bullet to max 5 • Class heading and ending (in appropriate place) • Constructor heading and ending (in appropriate place) • Declaring grid with correct dimensions (as private) • Declaring Sand as a constant • Nested loops covering dimensions (0 – 29 and 0 – 9) • Assigning Sand // '.' to each array element Example Python <pre>class IslandClass: def __init__(self): Sand = '.' self.__Grid = [[Sand for j in range(30)] for i in range(10)]</pre> Example Pascal <pre>type IslandClass = class private Grid : array[0..9, 0..29] of char; public constructor Create(); procedure HideTreasure(); procedure DigHole(x, y : integer); function GetSquare(x, y : integer) : char; end; constructor IslandClass.Create(); const Sand = '.'; var i, j : integer; begin for i := 0 to 9 do for j := 0 to 29 do Grid[i, j] := Sand; end; end;</pre>	5 1 1 1 1 + 1 1 1 1 1 1 1 1 1

Question	Answer	Marks
3(c)(ii)	1 mark per bullet • DisplayGrid header and ending, with two loops with correct limits • Calling Island.GetSquare with correct parameters inside iteration • Output an entire row in one line • Output a new line at the end of a row Example Python <pre>def DisplayGrid() : for i in range (10) : for j in range (30) : print(island.GetSquare(i, j), end='') print()</pre> Example Pascal <pre>procedure DisplayGrid(): var i, j : integer; begin for i := 0 to 9 do begin for j := 0 to 29 do write(island.GetSquare(i, j)); writeln; end; end;</pre> Example VB.NET <pre>Sub DisplayGrid() For i = 0 to 9 For j = 0 to 29 Console.Write(island.GetSquare(i, j)) Next Console.WriteLine() Next End Sub</pre>	4 1 1 + 1 1 1 1 + 1 1 1 1 + 1 1

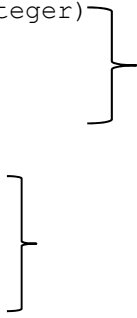
PUBLISHED

Question	Answer	Marks
3(d)	1 mark per bullet to max 5 • Method header and Declaring Treasure as a constant • Generating a random number for column • Generating a random number for row • Check whether treasure already at <u>generated</u> location • Repeatedly generate new coordinates in a loop • Assign Treasure to location Example Python <pre>def HideTreasure(self): Treasure = 'T' x = randint(0,9) y = randint(0,29) while self.__Grid[y][x] == Treasure: x = randint(0,9) y = randint(0,29) self.__Grid[y][x] = Treasure</pre> Example Pascal <pre>procedure IslandClass.HideTreasure(); const Treasure = 'T'; var x, y : integer; begin repeat x := Random(10); y := random(30); until Grid[x, y] <> Treasure; Grid[x, y] := Treasure; end;</pre>	Max 5 1 1 1 1+1 1 1 1 1 1+1 1

Question	Answer	Marks
	Example VB.NET <pre> Public Sub HideTreasure() Const Treasure = "T" Dim RandomNumber As New Random Dim x, y As Integer Do x = RandomNumber.Next(0, 10) y = RandomNumber.Next(0, 30) Loop Until Grid(x, y) <> Treasure Grid(x, y) = Treasure End Sub </pre>	1 1 1 1+1 1

PUBLISHED

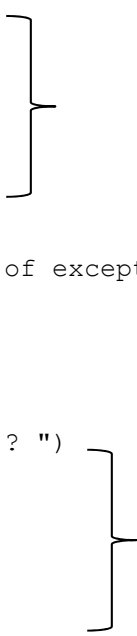
Question	Answer	Marks
3(e)(i)	1 mark per bullet • Method heading, with two parameters & Declaring constants for Treasure, Hole and FoundTreasure • Check if treasure at parameter locations • Set to FoundTreasure (X) and Set to Hole (O) Example Python <pre> def DigHole(self, x, y) : Treasure = 'T' Hole = 'O' Foundtreasure = 'X' if self.__Grid[x][y] == Treasure: self.__Grid[x][y] = Foundtreasure else : self.__Grid[x][y] = Hole return </pre> Example Pascal <pre> procedure IslandClass.DigHole(x, y : integer); const Treasure = 'T'; const Hole = 'O'; const Foundtreasure = 'X'; begin if Grid[x, y] = Treasure then Grid[x, y] := Foundtreasure else Grid[x, y] := Hole; end; </pre>	3 1 1 1 1 1 1

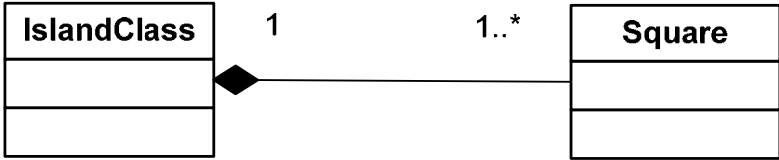
Question	Answer	Marks
	Example VB.NET <pre> Public Sub DigHole(x As Integer, y As Integer) Const Treasure = "T" Const Hole = "O" Const Foundtreasure = "X" If Grid(x, y) = Treasure Then Grid(x, y) = Foundtreasure Else Grid(x, y) = Hole End If End Sub </pre> 	1 1 1

PUBLISHED

Question	Answer	Marks
3(e)(ii)	1 mark per bullet to max 5 • Prompt to user for position down and across, read positions input as an IntegerValidation for position row – between 0 and 9 • Validation for position column- between 0 and 29 • Exception handling/pass for validation • Ask for repeated input until valid (for both row and column) • Call Island.DigHole method with the coordinates Example Python <pre>def StartDig() : Valid = False while not Valid : # validate down position try: x = int(input("position down <0 to 9> ? ")) if x >= 0 and x <= 9 : Valid = True except: Valid = False Valid = False while not Valid : # validate across position try : y = int(input("position across <0 to 29> ? ")) if y >= 0 and y <= 29 : Valid = True except : Valid = False island.DigHole(x, y) return</pre>	Max 5 1 1 1 1 1 1

Question	Answer	Marks
	Example Pascal <pre> procedure StartDig; var xString, yString : String; x, y : integer; begin Valid := False; repeat Write('position down <0 to 9>? '); ReadLn(xString); try x := StrToInt(xString); if (x >= 0) AND (x <= 9) then Valid := True; except Valid := False; until Valid; Valid := False; repeat Write(position across <0 to 29> ? '); ReadLn(yString); try y := StrToInt(yString); if (y >= 0) AND (y <= 29) then Valid := True; except Valid := False; until Valid; island.DigHole(x,y); end; </pre> 	1 1 1 1 1

Question	Answer	Marks
	Example VB.NET <pre> Sub StartDig() Dim x, y As Integer Dim Valid = False Do Console.WriteLine("Position down <0 to 9>? ") Try x = CInt(Console.ReadLine()) If (x >= 0) AND (x <= 9) Then Valid = True End If Catch Valid = False 'accept different types of exceptions End Try Loop Until Valid Valid = False Do Console.WriteLine("Position across <0 to 29> ? ") Try y = int(Console.ReadLine()) If (y >= 0) AND (y <= 29) Then Valid = True End IF Catch Valid = False End Try Loop until Valid island.DigHole(x, y) End Sub </pre> 	1 1 1 1 1
3(f)(i)	containment/aggregation	1

Question	Answer	Marks
3(f)(ii)	- IslandClass box and Square Box, with correct connection - One at IslandClass and one .. * at Square  <pre>classDiagram class IslandClass class Square IslandClass "1" *-- "1..*" Square</pre>	Max 2

QUESTION 5.

9608/41

Question	Answer	Marks
5(a)(i)	PERT / GANTT	1
5(a)(ii)	1 mark per bullet to max 3 For example: • Calculate total minimum time required for project • Identify milestones • Task dependencies • Provides the critical path analysis • Identify which tasks need to be prioritised • Determine when to begin specific tasks/stages • Identify slack time • Identify when resources need allocating • Identify tasks that can be completed in parallel	3
5(b)(i)	Integration	1
5(b)(ii)	Beta / acceptance	1

Question	Answer	Marks
6(a)	1 mark per bullet to max 6 • Declaring a class with the name animal • Declaring variables for across, down and score (all Integers) • ...as private/protected • Correct constructor header and ending • Randomly generating an across between 0–39 inc. in constructor • Randomly generating a down between 0–39 inc. in constructor • Initialising Score to zero in constructor • Correct get for Across • Correct set for Across	6

Question	Answer	Marks
6(a)	<pre>Example: VB Class Animal Private Across As Integer Private Down As Integer Private Score As Integer Function GetAcross() Return Across End Function Sub SetAcross(Value As Integer) Across = Value End Sub Sub New() Randomize() Across = randomnumber.Next(0, 40) Down = randomnumber.Next(0, 40) Score = 0 End Sub End Class</pre>	

Question	Answer	Marks
6(a)	or <pre> Class Animal Private Across As Integer Property _Across As Integer Get Return _Across End Get Set(Value As Integer) Across = Value End Set End Property Private Down As Integer Private _Score As Integer Sub New() Randomize() Across = randomnumber.Next(0, 40) Down = randomnumber.Next(0, 40) _Score = 0 End Sub End Class Example: Python class Animal : def __init__ (self) : x = random.randint(0,39) y = random.randint(0,39) self.Across = x self.Down = y self.Score = 0 def SetAcross(A) : self.Across = A def GetAcross() : return self.Across </pre>	

Question	Answer	Marks
6(a)	<pre> Example: Pascal type Animal = class private Across: integer; Down: integer; score: integer; public constructor init; procedure SetAcross(AcrossV: integer); function GetAcross(): integer; end; constructor Animal.init(); SetAcross(random(40)); SetDown (random(40)); SetScore (0); end; procedure Animal.SetAcross(AcrossV: integer); begin Across := AcrossV; end; function Animal.GetAcross(): integer; begin GetAcross := Across; end; </pre>	

Question	Answer	Marks
6(b)	1 mark per bullet to max 5 • constructor method heading and ending • Initialise all 40 by 40 elements of Grid as " or equivalent • Loop 5 times... • ...Creates a new instance of animal inside loop... • ...and adds it to array <code>AnimalList</code> • Call generate food and initialise <code>StepCounter</code> to 0 Example Python <pre>def __init__(self) : self.grid = [[' ' for i in range(40)] for j in range(40)] self.AnimalList = [] self.StepCounter = 0 for i in range(5) : newAnimal = Animal () self.AnimalList.append(newAnimal) self.GenerateFood()</pre> Example VB <pre>Sub New() For x = 0 To 39 For y = 0 To 39 grid(x, y) = "" Next Next For z = 0 To 4 AnimalList(z) = New Animal Next Call GenerateFood() End Sub</pre>	5

Question	Answer	Marks
6(b)	Example Pascal <pre> constructor Desert.init(); for x := 0 to 39 do begin for y := 0 to 39 do begin grid(x,y) = ""; end end end for x := 0 to 4 do begin AnimalList(x) = object (Animal); end GenerateFood(); end; </pre>	
6(c)(i)	1 mark per bullet: • Function header and ending taking one value as parameter • Check if coordinate = 0 (on lower bound) • ...generate random number (0 or 1) • Check if coordinate = 39 (on upper bound) • ...generate random number (–1 or 0) • Generate random number (e.g. –1, 0, 1) • Return the generated value	max 4

Question	Answer	Marks
6(c)(i)	Example VB <pre> Function GenerateDirection(ByRef coord As Integer) Dim lowerbound As Integer = -1 Dim upperbound As Integer = 1 If coord = 0 Then lowerbound = 0 ElseIf coord = 39 Then upperbound = 0 End If GenerateDirection = randomnumber.Next(lowerbound, upperbound) End Function </pre> Example Python <pre> def GenerateDirection(Coord) : lowerBound = -1 upperBound = 1 if Coord == 0 : lowerBound = 0 elif Coord == 39 : upperBound = 0 return random.randint(lowerBound, upperBound) </pre>	

Question	Answer	Marks
6(c)(i)	Example Pascal <pre>function GenerateDirection(coord : Integer): Integer; begin lowerbound = -1; upperbound = 1; if coord = 0 then lowerbound = 0; else if coord = 39 then upperbound = 0; GenerateDirection = random(39); end;</pre>	
6(c)(ii)	1 mark per bullet to max 4 • Procedure move header, no parameters • Calling GenerateDirection twice sending across and down as separate parameters • Add return value to Across • Add return value to Down • Check if the grid, at the (new) coordinates == "F" • ..if true, Call EatFood Example python <pre>def Move(self) : self.Across += GenerateChangeInCoordinate(self.Across) self.Down += GenerateChangeInCoordinate(self.Down) if grid[self.Across][self.Down] == 'F' : self.EatFood() return</pre>	4

Question	Answer	Marks
6(c)(ii)	Example VB <pre> Sub Move(ByRef thisAnimal As Animal) thisAnimal.across += GenerateChangeInCoordinate (thisAnimal.across) thisAnimal.down += GenerateChangeInCoordinate (thisAnimal.down) If thegrid._grid(thisAnimal.across, thisAnimal.down) = "F" Then Call EatFood() End If End Sub </pre> Example Pascal <pre> procedure Move(thisAnimal : Animal); begin thisAnimal.across = this.Animal.across + GenerateChangeInCoordinate (thisAnimal.across); thisAnimal.down = thisAnimal.down + GenerateChangeInCoordinate (thisAnimal.down); if (thisgrid.grid(thisAnimal.across, thisAnimal.down) = "F") then EatFood(); End; </pre>	
6(d)	1 mark per bullet to max 3 • Pre-compiled • Collection of Code/modules/routines • Each module performs a specific purpose/task • Each module can be linked/imported into the program	2

QUESTION 6.

9608/41

Question	Answer	Marks
2(a)	1 mark per bullet point to max 4: • declaration of type Book • Title, Author and ISBN as String • Fiction as Boolean • LastRead as Date For example: <pre> TYPE Book DECLARE Title : String DECLARE Author : String DECLARE ISBN : String DECLARE Fiction : Boolean DECLARE LastRead : Date ENDTYPE </pre>	4

PUBLISHED

Question	Answer	Marks
2(b)	1 mark per bullet point to max 4: • Function header • ... taking ISBN as parameter • Converting ISBN to integer • Calculating Hash (ISBN mod 2000 + 1) • Returning the calculated Hash Examples: Python: <pre>def Hash(ISBN): ISBNint = int(ISBN) Hash = (ISBNint % 2000) + 1</pre> VB.NET: <pre>Function Hash (ISBN As String) As Integer ISBNint = convert.ToInt32(ISBN) Hash = (ISBNint MOD 2000) + 1 End Function</pre> Pascal: <pre>function Hash(ISBN : String) : Integer begin ISBNint = StrToInt(ISBN) Hash = (ISBNint MOD 2000) + 1 end;</pre>	4

PUBLISHED

Question	Answer	Marks
2(c)	1 mark per bullet point to max 8: • Procedure FindBook declaration and prompt and input ISBN • Validate data input has 13 characters • ... and are all numeric • ..loop until valid • Call Hash() with input data and store return data • Open MyBooks.dat for reading as random file and close • Finding the record using return value Hash() • Get the data for the record • ...store in variable of type Book • ...output all the data for the record	8

Question	Answer	Marks
2(c)	Example: <pre> PROCEDURE FindBook() DECLARE BookInfo : Book REPEAT ISBN ← input("Enter the ISBN number") Valid ← TRUE Size ← LENGTH(ISBN) IF size <> 13 THEN Valid ← FALSE ELSE FOR i ← 1 to 13 IF NOT(MID(ISBN,i,1) >= '0' AND MID(ISBN,i,1)<= '9') THEN Valid ← FALSE ENDIF ENDFOR ENDIF UNTIL Valid Filename ← "myBooks.dat" OPENFILE Filename FOR RANDOM RecordLocation ← Hash(ISBN) SEEK FileName, RecordLocation GETRECORD Filename, BookInfo CLOSEFILE Filename OUTPUT (BookInfo.Title & " " & BookInfo.Author & " " & BookInfo.ISBN & " " & BookInfo.Fiction & " " & BookInfo.LastRead) ENDPROCEDURE </pre>	