



# Cambridge International AS & A Level

---

**COMPUTER SCIENCE****9618/41**

Paper 4 Practical

**May/June 2023**

MARK SCHEME

Maximum Mark: 75

**Published**

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the May/June 2023 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

---

This document consists of **38** printed pages.

**PUBLISHED****Generic Marking Principles**

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

**GENERIC MARKING PRINCIPLE 1:**

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

**GENERIC MARKING PRINCIPLE 2:**

Marks awarded are always **whole marks** (not half marks, or other fractions).

**GENERIC MARKING PRINCIPLE 3:**

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

**GENERIC MARKING PRINCIPLE 4:**

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

**PUBLISHED****GENERIC MARKING PRINCIPLE 5:**

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

**GENERIC MARKING PRINCIPLE 6:**

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

**PUBLISHED**

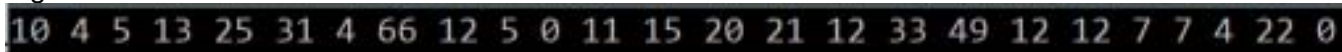
<b>Question</b>	<b>Answer</b>	<b>Marks</b>
1(a)(i)	<p>1 mark for</p> <ul style="list-style-type: none"><li>• 1D array with name <code>DataArray</code> (with 25 elements of type <code>Integer</code>)</li></ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static Integer[] dataArray = new Integer[25];</pre> <p><b>VB.NET</b></p> <pre>Dim dataArray(24) As Integer</pre> <p><b>Python</b></p> <pre>dataArray = [] #25 elements Integer</pre>	<b>1</b>

**PUBLISHED**

Question	Answer	Marks
1(a)(ii)	<p>1 mark each to max 4</p> <ul style="list-style-type: none"> <li>• Opening file <code>Data.txt</code> to read</li> <li>• Looping through all the 25/EOF ...</li> <li>• ... reading each line <b>and</b> storing/appending into array</li> <li>• Exception handling with appropriate output</li> <li>• Closing the file (in an appropriate place)</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>Integer Counter = 0; try{     Scanner Scanner1 = new Scanner(new File("Data.txt"));     while(Scanner1.hasNextLine()){         dataArray[Counter] = Integer.parseInt(Scanner1.next());         Counter++;     }     Scanner1.close(); }catch(FileNotFoundException ex){     System.out.println("No data file found"); }</pre> <p><b>VB.NET</b></p> <pre>try     Dim DataReader As New System.IO.StreamReader("Data.txt")     Dim X As Integer = 0     Do Until DataReader.EndOfStream         dataArray(X) = DataReader.ReadLine()         X = X + 1     Loop     DataReader.Close() Catch ex As Exception     Console.WriteLine("Invalid file") End Try</pre>	<b>4</b>

Question	Answer	Marks
1(a)(ii)	<b>Python</b> try: DataFile = open("Data.txt", 'r') for Line in DataFile: dataArray.append(int(Line)) DataFile.close() except IOError: print("Could not find file")	

Question	Answer	Marks
1(b)(i)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Procedure header (and close where appropriate) with (at least) one (integer array) parameter</li> <li>• Outputting all (25) array elements ...</li> <li>• ...on one line</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static void PrintArray(Integer[] dataArray){     String outputData;     for(Integer X = 0; X &lt; dataArray.length - 1; X++){         outputData = outputData + dataArray[X] + " ";     }     System.out.print(outputData); }</pre> <p><b>VB.NET</b></p> <pre>Sub PrintArray(dataArray)     Dim outputData As String = ""     For x = 0 To dataArray.length - 1         outputData = outputData &amp; dataArray(x) &amp; " "     Next     Console.WriteLine(outputData) End Sub</pre> <p><b>Python</b></p> <pre>def PrintArray(dataArray):     output = ""     for X in range(0, len(dataArray)):         output = output + str(dataArray[X]) + " "     print(output)</pre>	<b>3</b>

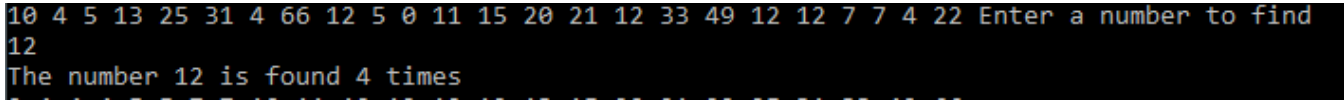
Question	Answer	Marks
1(b)(ii)	1 mark for calling <code>PrintArray</code> with the array as a parameter Example program code:  <b>Java</b> <code>PrintArray(DataArray);</code>  <b>VB.NET</b> <code>PrintArray(DataArray)</code>  <b>Python</b> <code>PrintArray(DataArray)</code>	<b>1</b>
1(b)(iii)	1 mark for screenshot  e.g. 	<b>1</b>



Question	Answer	Marks
1(c)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Function header (and close where appropriate) taking array <b>and</b> search value as parameters</li> <li>• Looping through each array element <b>and</b> keeping count of the number of times the <b>parameter</b> appears</li> <li>• Returning the <b>calculated</b> count value</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static Integer LinearSearch(Integer[] dataArray, Integer dataToFind){     Integer count = 0;     for(Integer x = 0; x &lt; dataArray.length - 1; x++){         if(dataArray[x] == dataToFind){             count++;         }     }     return count; }</pre> <p><b>VB.NET</b></p> <pre>Function LinearSearch(dataArray, dataToFind)     Dim count As Integer = 0     For x = 0 To dataArray.length - 1         If dataArray(x) = dataToFind Then             count = count + 1         End If     Next     Return count End Function</pre> <p><b>Python</b></p> <pre>def LinearSearch(dataArray, dataToFind):     count = 0     for X in range(0, len(dataArray)):         if(dataArray[X] == dataToFind):             count +=1     return count</pre>	<b>3</b>

Question	Answer	Marks
1(d)(i)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Prompt and reading input ...</li> <li>• ...with validation for whole number between 0 and 100 inclusive</li> <li>• Calling <code>LinearSearch()</code> with array and <b>valid</b> data input <b>and</b> storing/using return value</li> <li>• Output of the message with return value</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>System.out.println("Enter a number to find"); Integer DataToFind = -1; Scanner NewScanner = new Scanner(System.in); while(DataToFind &lt; 0    DataToFind &gt; 100){     DataToFind = Integer.parseInt(NewScanner.nextLine()); } Integer NumberTimes = LinearSearch(DataArray, DataToFind); System.out.println("The number " + DataToFind + " is found " + NumberTimes + " times");</pre> <p><b>VB.NET</b></p> <pre>Console.WriteLine("Enter a number to find ") Dim DataToFind As Integer = -1 Do Until DataToFind &gt;= 0 And DataToFind &lt;= 100     DataToFind = Console.ReadLine() Loop Dim NumberTimes = LinearSearch(DataArray, DataToFind) Console.WriteLine("The number " &amp; DataToFind &amp; " is found " &amp; NumberTimes &amp; " times.")</pre> <p><b>Python</b></p> <pre>DataToFind = int(input("Enter a number to find ")) while DataToFind &lt; 0 or DataToFind &gt; 100:     DataToFind = int(input("Enter a number to find ")) NumberTimes = LinearSearch(DataArray, DataToFind) print("The number", DataToFind, "is found", NumberTimes, "times")</pre>	<b>4</b>

**PUBLISHED**

Question	Answer	Marks
1(d)(ii)	1 mark for screenshot e.g.  <pre>10 4 5 13 25 31 4 66 12 5 0 11 15 20 21 12 33 49 12 12 7 7 4 22 Enter a number to find 12 The number 12 is found 4 times</pre>	<b>1</b>

**PUBLISHED**

Question	Answer	Marks
2(a)(i)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Class header (and close where appropriate)</li> <li>• 5 (private) attribute declarations including data types</li> <li>• Constructor header (and close where appropriate) taking 3 parameters (min)</li> <li>• Assigning ID, MaxSpeed and IncreaseAmount to parameters</li> <li>• Assigning CurrentSpeed and HorizontalPosition to 0</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Class Vehicle     Private ID As String     Private MaxSpeed As Integer     Private CurrentSpeed As Integer     Private IncreaseAmount As Integer     Private HorizontalPosition As Integer      Sub New(IDP, MaxSpeedP, IncreaseAmountP)         ID = IDP         MaxSpeed = MaxSpeedP         CurrentSpeed = 0         IncreaseAmount = IncreaseAmountP         HorizontalPosition = 0     End Sub End Class</pre> <p><b>Java</b></p> <pre>class Vehicle{     private String ID;     private Integer MaxSpeed;     private Integer CurrentSpeed;     private Integer IncreaseAmount;     private Integer HorizontalPosition;</pre>	<b>5</b>

**PUBLISHED**

Question	Answer	Marks
2(a)(i)	<pre> public Vehicle(String IDP, Integer MaxSpeedP, Integer IncreaseAmountP){     ID = IDP;     MaxSpeed = MaxSpeedP;     IncreaseAmount = IncreaseAmountP;     CurrentSpeed = 0;     HorizontalPosition = 0; } }  <b>Python</b> class Vehicle:     #self.__ID string     #self.__MaxSpeed integer     #self.__CurrentSpeed integer     #self.__IncreaseAmount integer     #self.__HorizontalPosition      def __init__(self, IDP, MaxSpeedP, IncreaseAmountP):         self.__ID = IDP         self.__MaxSpeed = MaxSpeedP         self.__IncreaseAmount = IncreaseAmountP         self.__CurrentSpeed = 0         self.__HorizontalPosition = 0 </pre>	

**PUBLISHED**

Question	Answer	Marks
2(a)(ii)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• 1 get function header (and end where appropriate) with no parameter ...</li> <li>• ...returning attribute (without overwriting)</li> <li>• 3 further <b>correct</b> get methods</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Function GetCurrentSpeed()     Return CurrentSpeed End Function Function GetIncreaseAmount()     Return IncreaseAmount End Function Function GetHorizontalPosition()     Return HorizontalPosition End Function Function GetMaxSpeed()     Return MaxSpeed End Function</pre> <p><b>Java</b></p> <pre>public Integer GetCurrentSpeed(){     return CurrentSpeed; } public Integer GetIncreaseAmount(){     return IncreaseAmount; } public Integer GetHorizontalPosition(){     return HorizontalPosition; } public Integer GetMaxSpeed(){     return MaxSpeed; }</pre>	<b>3</b>

**PUBLISHED**

<b>Question</b>	<b>Answer</b>	<b>Marks</b>
2(a)(ii)	<b>Python</b> <pre>def GetCurrentSpeed(self):     return self.__CurrentSpeed def GetIncreaseAmount(self):     return self.__IncreaseAmount def GetHorizontalPosition(self):     return self.__HorizontalPosition def GetMaxSpeed(self):     return self.__MaxSpeed</pre>	

**PUBLISHED**

Question	Answer	Marks
2(a)(iii)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• 1 set procedure (and end where appropriate) taking parameter ...</li> <li>• ... assigns parameter to the attribute (without overriding)</li> <li>• Second <b>correct</b> set method</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Sub SetCurrentSpeed(CSp)     CurrentSpeed = CSp End Sub Sub SetHorizontalPosition(HPP)     HorizontalPosition = HPP End Sub</pre> <p><b>Java</b></p> <pre>public void SetCurrentSpeed(Integer CSP){     CurrentSpeed = CSP; } public void SetHorizontalPosition(Integer HPP){     HorizontalPosition = HPP; }</pre> <p><b>Python</b></p> <pre>def SetCurrentSpeed(self, CSP):     self.__CurrentSpeed = CSP def SetHorizontalPosition(self, HPP):     self.__HorizontalPosition = HPP</pre>	<b>3</b>



**PUBLISHED**

Question	Answer	Marks
2(a)(iv)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Method header (and close where appropriate) with no parameter <b>and</b> adding IncreaseAmount to CurrentSpeed</li> <li>• Checking if MaxSpeed is exceeded <b>and</b> limiting to max speed (remove increase or assign maximum)</li> <li>• Adding updated CurrentSpeed to HorizontalPosition in all cases (whether MaxSpeed is exceeded or not)</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Sub IncreaseSpeed()     CurrentSpeed = CurrentSpeed + IncreaseAmount     If CurrentSpeed &gt; MaxSpeed Then         CurrentSpeed = MaxSpeed     End If     HorizontalPosition = HorizontalPosition + CurrentSpeed End Sub</pre> <p><b>Java</b></p> <pre>public void IncreaseSpeed(){     CurrentSpeed = CurrentSpeed + IncreaseAmount;     if(CurrentSpeed &gt; MaxSpeed){         CurrentSpeed = MaxSpeed;     }     HorizontalPosition = HorizontalPosition + CurrentSpeed; }</pre> <p><b>Python</b></p> <pre>def IncreaseSpeed(self):     self.__CurrentSpeed = self.__CurrentSpeed + self.__IncreaseAmount     if(self.__CurrentSpeed &gt; self.__MaxSpeed):         self.__CurrentSpeed = self.__MaxSpeed     self.__HorizontalPosition = self.__HorizontalPosition + self.__CurrentSpeed</pre>	<b>3</b>

**PUBLISHED**

Question	Answer	Marks
2(b)(i)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Class header (and end where appropriate) inheriting from Vehicle</li> <li>• 3 (private) attribute declarations with data types</li> <li>• Constructor (and end where appropriate) with (min) 5 parameters</li> <li>• Calling parent constructor with appropriate parameters</li> <li>• Initialising <code>VerticalPosition</code> to 0 and <code>VerticalChange</code> and <code>MaxHeight</code> to attributes</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre> Class Helicopter     Inherits Vehicle     Private VerticalPosition As Integer     Private VerticalChange As Integer     Private MaxHeight As Integer      Sub New(IDP, MaxSpeedP, IncreaseAmountP, VertChangeP, MaxHeightP)         MyBase.New(IDP, MaxSpeedP, IncreaseAmountP)         VerticalPosition = 0         VerticalChange = VertChangeP         MaxHeight = MaxHeightP     End Sub End Class </pre> <p><b>Java</b></p> <pre> class Helicopter extends Vehicle{     private Integer VerticalPosition;     private Integer VerticalChange;     private Integer MaxHeight;      public Helicopter(String IDP, Integer MaxSpeedP, Integer IncreaseAmountP, Integer         VertChangeP, Integer MaxHeightP){ </pre>	5

**PUBLISHED**

Question	Answer	Marks
2(b)(i)	<pre> super(IDP, MaxSpeedP, IncreaseAmountP); VerticalPosition = 0; VerticalChange = VertChangeP; MaxHeight = MaxHeightP; }}  <b>Python</b> class Helicopter(Vehicle):     #VerticalPosition Integer     #VerticalChange Integer     #MaxHeight Integer      def __init__(self, IDP, MaxSpeedP, IncreaseAmountP, VertChangeP, MaxHeightP):         Vehicle.__init__(self, IDP, MaxSpeedP, IncreaseAmountP)         self.__VerticalPosition = 0         self.__VerticalChange = VertChangeP         self.__MaxHeight = MaxHeightP </pre>	

Question	Answer	Marks
2(b)(ii)	<p>1 mark each to max 4</p> <ul style="list-style-type: none"> <li>• Method header (overriding where required) with no parameter</li> <li>• Adding vertical change to vertical position ...</li> <li>• ...limiting to maximum height</li> <li>• Repeating/calling/using the code from original for horizontal increase (in every case)</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre> Overrides Sub IncreaseSpeed()     VerticalPosition = VerticalPosition + VerticalChange     If VerticalPosition &gt; MaxHeight Then         VerticalPosition = MaxHeight     End If     Me.SetCurrentSpeed(GetCurrentSpeed() + GetIncreaseAmount())     If Me.GetCurrentSpeed() &gt; Me.GetMaxSpeed() Then         Me.SetCurrentSpeed(Me.GetMaxSpeed())     End If     Me.SetHorizontalPosition(Me.GetHorizontalPosition() + Me.GetCurrentSpeed()) End Sub </pre> <p><b>Java</b></p> <pre> public void IncreaseSpeed(){     VerticalPosition = VerticalPosition + VerticalChange;     if(VerticalPosition &gt; MaxHeight){         VerticalPosition = MaxHeight;     }     super.SetCurrentSpeed(super.GetCurrentSpeed() + super.GetIncreaseAmount());     if(super.GetCurrentSpeed() &gt; super.GetMaxSpeed()){         super.SetCurrentSpeed(super.GetMaxSpeed());     }     super.SetHorizontalPosition(super.GetHorizontalPosition() + super.GetCurrentSpeed()); } </pre>	4

**PUBLISHED**

Question	Answer	Marks
2(b)(ii)	<p><b>Python</b></p> <pre>def IncreaseSpeed(self):     self.__VerticalPosition = self.__VerticalPosition + self.__VerticalChange     if(self.__VerticalPosition &gt; self.__MaxHeight):         self.__VerticalPosition = MaxHeight     Vehicle.SetCurrentSpeed(self, Vehicle.GetCurrentSpeed(self) + Vehicle.GetIncreaseAmount(self))     if(Vehicle.GetCurrentSpeed(self) &gt; Vehicle.GetMaxSpeed(self)):         Vehicle.SetCurrentSpeed(self, Vehicle.GetMaxSpeed(self));     Vehicle.SetHorizontalPosition(self, Vehicle.GetHorizontalPosition(self) + Vehicle.GetCurrentSpeed(self))</pre>	

Question	Answer	Marks
2(c)	<p>1 mark each to max 3</p> <ul style="list-style-type: none"> <li>• Suitable method/procedure heading (and end where appropriate) <b>and</b> outputting horizontal position <b>and</b> current speed in an appropriate message</li> <li>• Checking if object is a Vehicle or Helicopter // overriding methods in each class for output // one method in each class // try except ...</li> <li>• ...outputting vertical position only if helicopter with appropriate message</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Sub OutputCurrentPosition(ObjectToOutput)     Console.WriteLine("Current position = " &amp; ObjectToOutput.GetHorizontalPosition())     Console.WriteLine("Current speed = " &amp; ObjectToOutput.GetCurrentSpeed())     If TypeOf ObjectToOutput Is Helicopter Then         Console.WriteLine("Current vertical position = " &amp; ObjectToOutput.GetVerticalPosition())     End If End Sub</pre> <p><b>Java</b></p> <pre>public void OutputCurrentPosition(){     System.out.println("Current position = " + HorizontalPosition);     System.out.println("Current speed = " + CurrentSpeed); }  public void OutputCurrentPosition(){     System.out.println("Current position = " +super.GetHorizontalPosition());     System.out.println("Current speed = " + super.GetCurrentSpeed());     System.out.println("Current vertical position = " + VerticalPosition); }</pre> <p><b>Python</b></p> <pre>def OutputCurrentPosition(self):     print("Current position = ", self.__HorizontalPosition)     print("Current speed = ", self.__CurrentSpeed)</pre>	<b>3</b>

**PUBLISHED**

<b>Question</b>	<b>Answer</b>	<b>Marks</b>
2(c)	<pre>def OutputCurrentPosition(self):     print("Current position = ", Vehicle.GetHorizontalPosition(self))     print("Current speed = ", Vehicle.GetCurrentSpeed(self))     print("Current verticalposition = ", self.__VerticalPosition)</pre>	

**PUBLISHED**

Question	Answer	Marks
2(d)(i)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Instantiating an object of type <code>Vehicle</code> with correct parameters ("Tiger", 100, 20)</li> <li>• Instantiating an object of type <code>Helicopter</code> with correct parameters ("Lion", 350, 40, 3, 100)</li> <li>• Calling <code>IncreaseSpeed()</code> twice for the car</li> <li>• Calling <code>IncreaseSpeed()</code> twice for the helicopter</li> <li>• Calling the output for both objects</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Sub Main()     Dim Car As Vehicle     Car = New Vehicle("Tiger", 100, 20)     Dim Helil As Helicopter     Helil = New Helicopter("Lion", 350, 40, 3, 100)     Car.IncreaseSpeed()     Car.IncreaseSpeed()     OutputCurrentPosition(Car)     Console.WriteLine("")     Helil.IncreaseSpeed()     Helil.IncreaseSpeed()     OutputCurrentPosition(Helil) End Sub</pre> <p><b>Java</b></p> <pre>public static void main(String args[]){     Vehicle Car = new Vehicle("Tiger", 100, 20);     Helicopter Helil = new Helicopter("Lion", 350, 40, 3, 100);     Car.IncreaseSpeed();     Car.IncreaseSpeed();     Car.OutputCurrentPosition();     System.out.println("");     Helil.IncreaseSpeed();     Helil.IncreaseSpeed();     Helil.OutputCurrentPosition(); }</pre>	<b>5</b>



**PUBLISHED**

Question	Answer	Marks
2(d)(i)	<b>Python</b> <pre>#main Car = Vehicle("Tiger", 100, 20) Helil = Helicopter("Lion", 350, 40, 3, 100) Car.IncreaseSpeed() Car.IncreaseSpeed() Car.OutputCurrentPosition() print("") Helil.IncreaseSpeed() Helil.IncreaseSpeed() Helil.OutputCurrentPosition()</pre>	
2(d)(ii)	Screenshot of results e.g. <pre>Current position = 60 Current speed = 40  Current position = 120 Current speed = 80 Current verticalposition = 6</pre>	<b>1</b>

**PUBLISHED**

Question	Answer	Marks
3(a)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• (Global) Animal array (with 20 string elements)</li> <li>• (Global) Colour array (with 10 string elements)</li> <li>• (Global) AnimalTopPointer and ColourTopPointer initialised to 0</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static String[] Animal = new String[20]; public static String[] Colour = new String[10]; public static Integer AnimalTopPointer = 0; public static Integer ColourTopPointer = 0;</pre> <p><b>VB.NET</b></p> <pre>Dim Animal(0 to 19) As String Dim Colour(0 to 9) As String Dim AnimalTopPointer As Integer = 0 Dim ColourTopPointer As Integer = 0</pre> <p><b>Python</b></p> <pre>Animal = [] #20 elements Colour = [] #10 elements global AnimalTopPointer global ColourTopPointer AnimalTopPointer = 0 ColourTopPointer = 0</pre>	<b>3</b>

**PUBLISHED**

Question	Answer	Marks
3(b)(i)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Function header (and close where appropriate) with parameter, checking if full (AnimalTopPointer = 20) <b>and</b> returning false</li> <li>• If not full, inserting parameter value into AnimalTopPointer</li> <li>• ...incrementing pointer <b>and</b> returning true</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static Boolean PushAnimal(String DataToPush){     if(AnimalTopPointer == 20){         return false;     }else{         Animal[AnimalTopPointer] = DataToPush;         AnimalTopPointer++;         return true;     } }</pre> <p><b>VB.NET</b></p> <pre>Function PushAnimal(DataToPush)     If AnimalTopPointer = 20 Then         Return False     Else         Animal(AnimalTopPointer) = DataToPush         AnimalTopPointer = AnimalTopPointer + 1         Return True     End If End Function</pre> <p><b>Python</b></p> <pre>def PushAnimal(DataToPush):     global AnimalTopPointer     global ColourTopPointer     if AnimalTopPointer == 20:         return False</pre>	<b>3</b>

Question	Answer	Marks
3(b)(i)	<pre> else:     Animal.append(DataToPush)     AnimalTopPointer +=1     return True </pre>	
3(b)(ii)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Procedure header (and end where appropriate) with no parameter, checking if empty (<code>AnimalTopPointer = 0</code>) <b>and</b> returning empty string</li> <li>• If not empty returning the top data item (<code>AnimalTopPointer-1</code>)</li> <li>• ... and decrementing <code>AnimalTopPointer</code></li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre> public static String PopAnimal(){     String ReturnData;     if(AnimalTopPointer == 0){         return "";     }else{         ReturnData = Animal[AnimalTopPointer - 1];         AnimalTopPointer--;         return ReturnData;     } } </pre> <p><b>VB.NET</b></p> <pre> Function PopAnimal()     Dim ReturnData As String     If AnimalTopPointer = 0 Then         Return ""     Else         ReturnData = Animal(AnimalTopPointer - 1)         AnimalTopPointer = AnimalTopPointer - 1         Return ReturnData     End If End Function </pre>	<b>3</b>

Question	Answer	Marks
3(b)(ii)	<b>Python</b> <pre>def PopAnimal():     global AnimalTopPointer     global ColourTopPointer     if AnimalTopPointer == 0:         return ""     else:         ReturnData = Animal[AnimalTopPointer - 1]         AnimalTopPointer -=1         return ReturnData</pre>	

Question	Answer	Marks
3(b)(iii)	<p>1 mark</p> <ul style="list-style-type: none"> <li>• Procedure header (and close where appropriate) <b>and</b> opening correct file for read</li> <li>• Looping until end of file // looping until all animal names read in // looping 8 times</li> <li>• Calling <code>PushAnimal()</code> with each line read from file (for all lines)</li> <li>• Closing the file</li> <li>• Exception handling with appropriate error message</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>private static void ReadData(){     try{         Scanner Scanner1 = new Scanner(new File("AnimalData.txt"));         while(Scanner1.hasNextLine()){             PushAnimal(Scanner1.next());         }         Scanner1.close();     }catch(FileNotFoundException ex){         System.out.println("No Animal file found");     } }</pre> <p><b>VB.NET</b></p> <pre>Sub ReadData()     try         Dim AnimalFile As String = "AnimalData.txt"         Dim AnimalFileReader As New System.IO.StreamReader(AnimalFile)         Do Until AnimalFileReader.EndOfStream             PushAnimal(AnimalFileReader.ReadLine())         Loop         AnimalFileReader.Close()     Catch ex As Exception         Console.WriteLine("Invalid file")     End Try End Sub</pre>	<b>5</b>

Question	Answer	Marks
3(b)(iii)	<b>Python</b> <pre>def ReadData():     try:         global AnimalTopPointer         global ColourTopPointer         AnimalFile = open("AnimalData.txt", 'r')         for Line in AnimalFile:             PushAnimal(Line)         AnimalFile.close()     except IOError:         print("Could not find file")</pre>	

Question	Answer	Marks
3(b)(iv)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• PushColour function</li> <li>• PopColour function</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static Boolean PushColour(String DataToPush){     if(ColourTopPointer == 10){         return false;     }else{         Colour[ColourTopPointer] = DataToPush;         ColourTopPointer++;         return true;     } }  public static String PopColour(){     String ReturnData;     if(ColourTopPointer == 0){         return "";     }else{         ReturnData = Colour[ColourTopPointer - 1];         ColourTopPointer--;         return ReturnData;     } }</pre> <p><b>VB.NET</b></p> <pre>Function PushColour(DataToPush)     If ColourTopPointer = 10 Then         Return False     Else         Colour(ColourTopPointer) = DataToPush         ColourTopPointer = ColourTopPointer + 1         Return True     End If</pre>	2



**PUBLISHED**

Question	Answer	Marks
3(b)(iv)	<pre> End Function Function PopColour()   Dim ReturnData As String   If ColourTopPointer = 0 Then     Return ""   Else     ReturnData = Colour(ColourTopPointer - 1)     ColourTopPointer = ColourTopPointer - 1     Return ReturnData   End If End Function  <b>Python</b> def PushColour(DataToPush):     global AnimalTopPointer     global ColourTopPointer     if ColourTopPointer == 10:         return False     else:         Colour.append(DataToPush)         ColourTopPointer +=1         return True  def PopColour():     global AnimalTopPointer     global ColourTopPointer     if ColourTopPointer == 0:         return ""     else:         ReturnData = Colour[ColourTopPointer - 1]         ColourTopPointer -=1         return ReturnData </pre>	

Question	Answer	Marks
3(b)(v)	<p>1 mark each</p> <ul style="list-style-type: none"> <li>• Opening <code>ColourData.txt</code> to read, reading until EOF, closing file and exception handling</li> <li>• Using <code>PushColour()</code> to store each item read from the file <b>for all lines</b></li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>private static void ReadData(){     try{         Scanner Scanner1 = new Scanner(new File("AnimalData.txt"));         while(Scanner1.hasNextLine()){             PushAnimal(Scanner1.next());         }         Scanner1.close();     }catch(FileNotFoundException ex){         System.out.println("No Animal file found");     }      try{         Scanner Scanner2 = new Scanner(new File("ColourData.txt"));         while(Scanner2.hasNextLine()){             PushColour(Scanner2.next());         }         Scanner2.close();     }catch(FileNotFoundException ex){         System.out.println("No Colour file found");     } }</pre> <p><b>VB.NET</b></p> <pre>Sub ReadData()     try         Dim AnimalFile As String = "AnimalData.txt"         Dim AnimalFileReader As New System.IO.StreamReader(AnimalFile)         Do Until AnimalFileReader.EndOfStream             PushAnimal(AnimalFileReader.ReadLine())         Loop     Catch     End Try End Sub</pre>	<b>2</b>

Question	Answer	Marks
3(b)(v)	<pre> Loop AnimalFileReader.Close() Dim ColourFile As String = "ColourData.txt" Dim ColourFileReader As New System.IO.StreamReader(ColourFile) Do Until ColourFileReader.EndOfStream     PushColour(ColourFileReader.ReadLine()) Loop ColourFileReader.Close() Catch ex As Exception     Console.WriteLine("Invalid file") End Try  End Sub  <b>Python</b> def ReadData():     try:         global AnimalTopPointer         global ColourTopPointer         AnimalFile = open("AnimalData.txt", 'r')         for Line in AnimalFile:             PushAnimal(Line)         AnimalFile.close()          ColourFile = open("ColourData.txt", 'r')         for Line in ColourFile:             PushColour(Line)         ColourFile.close()     except IOError:         print("Could not find file") </pre>	

Question	Answer	Marks
3(c)	<p>1 mark each to max 5</p> <ul style="list-style-type: none"> <li>• Procedure heading (and close where appropriate) <b>and</b> outputting the colour and animal using PopColour() <b>and</b> PopAnimal() (only if both are successfully popped)</li> <li>• Checking if no colour <b>and</b> outputting "No colour" ...</li> <li>• ...pushing the removed animal back onto the stack</li> <li>• Checking if no animal <b>and</b> outputting "No animal" ...</li> <li>• ...pushing the removed colour back onto the stack</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static void OutputItem(){     String ColourReturned = PopColour();     String AnimalReturned = PopAnimal();     if(ColourReturned.equals("")){         System.out.println("No colour");         PushAnimal(AnimalReturned);     }else{         if(AnimalReturned.equals("")){             System.out.println("No animal");             PushColour(ColourReturned);         }else{             System.out.println("A " + ColourReturned + " " + AnimalReturned);         }     } }</pre> <p><b>VB.NET</b></p> <pre>Sub OutputItem()     Dim ColourReturned As String = PopColour()     Dim Animalreturned As String = PopAnimal()     If ColourReturned = "" Then         Console.WriteLine("No colour")         PushAnimal(AnimalReturned)</pre>	<b>5</b>

Question	Answer	Marks
3(c)	<pre> Else   If AnimalReturned = "" Then     Console.WriteLine("No animal")     PushColour(ColourReturned)   Else     Console.WriteLine("A " &amp; ColourReturned &amp; " " &amp; AnimalReturned)   End If End If End Sub  <b>Python</b> def OutputItem():   global AnimalTopPointer   global ColourTopPointer   ColourReturned = PopColour()   AnimalReturned = PopAnimal()   if ColourReturned == "":     print("No colour")     PushAnimal(AnimalReturned)   else:     if AnimalReturned == "":       print("No animal")       PushColour(ColourReturned)     else:       print(ColourReturned, AnimalReturned) </pre>	

Question	Answer	Marks
3(d)(i)	<p>1 mark for</p> <ul style="list-style-type: none"> <li>• Calling ReadData() and calling OutputItem() 4 times</li> </ul> <p>Example program code:</p> <p><b>Java</b></p> <pre>public static void main(String args[]){     ReadData();     OutputItem();     OutputItem();     OutputItem();     OutputItem(); }</pre> <p><b>VB.NET</b></p> <pre>Sub Main()     ReadData()     OutputItem()     OutputItem()     OutputItem()     OutputItem() End Sub</pre> <p><b>Python</b></p> <pre>ReadData() OutputItem() OutputItem() OutputItem() OutputItem()</pre>	<b>1</b>
3(d)(ii)	<p>1 mark for output e.g.</p> <pre>pink hamster blue elephant white eagle yellow rabbit</pre>	<b>1</b>