



## Cambridge International AS & A Level

---

**COMPUTER SCIENCE**

**9618/42**

Paper 42 Practical

**May/June 2022**

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 2022 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 **36** 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.



**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.



| Question | Answer  | Marks |
|----------|---|-------|
| 1(a)     | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• declaring array StackData and pointer StackPointer as (global data structures)</li> <li>• StackData has 10 integer elements</li> <li>• StackPointer initialised to 0</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Dim StackData(9) As Integer Dim StackPointer As Integer Sub Main()     StackPointer = 0 end Sub</pre> <p><b>Python</b></p> <pre>global StackData #integer global StackPointer StackData = [0,0,0,0,0,0,0,0,0,0] #integer StackPointer = 0</pre> <p><b>Java</b></p> <pre>import java.util.Scanner; class Question1{     public static Integer[] StackData;     public static Integer StackPointer;     public static void main(String args[]){         StackData = new Integer[10];         StackPointer = 0;     } }</pre> | 3     |



| Question | Answer  | Marks    |
|----------|---|----------|
| 1(b)     | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• procedure header with sensible identifier (and end where appropriate)</li> <li>• outputting StackPointer</li> <li>• outputting all <b>10</b> elements in array</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Sub PrintArray()     Console.WriteLine(StackPointer)     For x = 0 To 9         Console.WriteLine(StackData(x))     Next End Sub</pre> <p><b>Python</b></p> <pre>def PrintArray():     global StackData     global StackPointer     print(StackPointer)     for x in range (0, 10):         print(StackData[x])</pre> <p><b>Java</b></p> <pre>public static void PrintArray(){     System.out.println(StackPointer);     for(int x = 0; x &lt; 10 ;x++){         System.out.println(StackData[x]);     } }</pre> | <b>3</b> |



| Question | Answer   | Marks    |
|----------|--|----------|
| 1(c)     | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• function <code>Push()</code> taking an integer parameter</li> <li>• checking if stack is full ...</li> <li>• ...and returning <code>FALSE</code></li> <li>• (if not full) <b>storing parameter</b> to stack at <code>StackPointer</code> ...</li> <li>• ...incrementing <code>StackPointer</code></li> <li>• ...returning <code>TRUE</code></li> </ul> <p>Example program code:</p> <p><b>VB.Net</b></p> <pre>Function Push(DataToPush)     If StackPointer = 10 Then         Return False     Else         StackData(StackPointer) = DataToPush         StackPointer = StackPointer + 1         Return True     End If End Function</pre> <p><b>Python</b></p> <pre>def Push(DataToPush):     global StackData     global StackPointer     if StackPointer == 10:         return False     else:         StackData[StackPointer] = DataToPush         StackPointer = StackPointer + 1         return True</pre> | <b>6</b> |



| Question | Answer   | Marks |
|----------|--|-------|
| 1(c)     | <b>Java</b><br><pre>public static Boolean Push(Integer DataToPush){     if(StackPointer == 10){         return false;     }else{         StackData[StackPointer] = DataToPush;         StackPointer = StackPointer + 1;         return true;     } }</pre> |       |



| Question | Answer   | Marks    |
|----------|--|----------|
| 1(d)(i)  | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• Inputting 11 numbers ...</li> <li>• ...calling <code>Push()</code> with each number input as a parameter ...</li> <li>• ...outputting appropriate message if <code>TRUE</code> returned</li> <li>• ...outputting appropriate message if <code>FALSE</code> returned</li> </ul><br><ul style="list-style-type: none"> <li>• Calling their output procedure <b>after</b> all <b>11</b> additions</li> </ul> <p>Example program code:<br/><b>VB.NET</b></p> <pre> Sub Main()     StackPointer = 0     Dim TempNumber As Integer     For x = 0 To 10         Console.WriteLine("Enter a number")         TempNumber = Console.ReadLine()          If Push(TempNumber) Then             Console.WriteLine("Stored")         Else             Console.WriteLine("Stack full")         End If     Next     PrintArray()     Console.ReadLine() End Sub </pre> | <b>5</b> |





| Question | Answer  | Marks |
|----------|---|-------|
| 1(d)(i)  | <p><b>Python</b></p> <pre>#main StackPointer = 0 StackData = [0,0,0,0,0,0,0,0,0,0] for x in range(0, 11):     TempNumber = int(input("Enter a number"))      if Push(TempNumber) == True:         print("Stored")     else:         print("Stack full") PrintArray()</pre> <p><b>Java</b></p> <pre>public static void main(String[] args){     StackData = new Integer[10];     StackPointer = 0;     Integer TempNumber = 0;     for(int x = 0; x &lt; 10; x++){         System.out.println("Enter a number");         Scanner scanner = new Scanner(System.in);         TempNumber = Integer.parseInt(scanner.nextLine());         if(Push(TempNumber)){             System.out.println("Stored");         }else{             System.out.println("Stack full");         }     }     PrintArray(); }</pre> |       |

| Question | Answer   | Marks    |
|----------|--|----------|
| 1(d)(ii) | <p>1 mark for inputting all 11 numbers, message for first 10 saying added (11 to 20), message stating 11th number stating stack full. Full array contents output (11 12 13 14 15 16 17 18 19 20).<br/>e.g.</p> <pre> Enter a number 11 stored Enter a number 12 stored Enter a number 13 stored Enter a number 14 stored Enter a number 15 stored Enter a number 16 stored Enter a number 17 stored Enter a number 18 stored Enter a number 19 stored Enter a number 20 stored stack full pointer 18 output 11 12 13 14 15 16 17 18 19 20 </pre> | <b>1</b> |

| Question | Answer   | Marks    |
|----------|--|----------|
| 1(e)(i)  | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• <code>Pop()</code> function header (and close where appropriate) <b>and</b> returning a number in all possible situations</li> <li>• checking if stack is empty (<code>StackPointer</code> is 0) <b>and</b> returning -1</li> <li>• (otherwise) <b>accessing</b> the item at the top of the stack ...</li> <li>• ...decrementing the stack pointer</li> <li>• ...returning the item removed</li> </ul> <p>Example Program code:</p> <p><b>VB.NET</b></p> <pre>Function Pop()     Dim ReturnData As Integer     If StackPointer = 0 Then         Return -1     Else         ReturnData = StackData(StackPointer - 1)         StackPointer = StackPointer - 1         Return ReturnData     End If End Function</pre> <p><b>Python</b></p> <pre>def Pop():     global StackData     global StackPointer     if StackPointer == 0:         return -1     else:         ReturnData = StackData[StackPointer - 1]         StackPointer = StackPointer - 1         return ReturnData</pre> | <b>5</b> |



| Question | Answer  | Marks    |
|----------|---|----------|
| 1(e)(i)  | <p><b>Java</b></p> <pre>public static Integer Pop(){     Integer ReturnData = 0;     if(StackPointer == 0){         return -1;     }else{         ReturnData = StackData[StackPointer - 1];         StackPointer = StackPointer - 1;         return ReturnData;     } }</pre>   |          |
| 1(e)(ii) | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• output of before removed with 11 inputs</li> <li>• output of stack (after, this could be 11–20, 11–18, or 11–18 then 'null' 'null's)</li> </ul> <p>e.g.</p> <pre>Enter a number11 Stored Enter a number12 Stored Enter a number13 Stored Enter a number14 Stored Enter a number15 Stored Enter a number16 Stored Enter a number17 Stored Enter a number18 Stored Enter a number19 Stored Enter a number20 Stored Enter a number21 Stack full g Output 11 12 13 14 15 16 17 18 19 20</pre> | <b>2</b> |

| Question | Answer  | Marks |
|----------|---|-------|
| 2(a)     | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• in main local 2D array declared ...</li> <li>• ... with 10 × 10 integer elements</li> <li>• initialising <b>all</b> array elements to a number...</li> <li>• ...that is random between 1 and 100 (allow inclusive or exclusive)</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Sub Main()     Dim Random As New Random     Dim ArrayData(10, 10) As Integer     For x = 0 To 9         For y = 0 To 9             ArrayData(x, y) = Random.Next(1, 100)         Next     Next     Console.ReadLine() End Sub</pre> <p><b>Python</b></p> <pre>import random #main ArrayData= [[0]*10 for i in range(10)] #integer for x in range(0, 10):     for y in range(0,10):         ArrayData[x][y] = random.randint(1, 100)</pre> | 4     |



| Question | Answer  | Marks |
|----------|---|-------|
| 2(a)     | <b>Java</b><br>import java.util.Scanner;<br>import java.util.Random;<br><br>class Question2{<br>public static Integer[][] ArrayData;<br>public static void main(String args[]){<br>Random Rand = new Random();<br>ArrayData = new Integer[10][10];<br>for(int x=0; x < 10; x++){<br>for(int y = 0; y < 10; y++){<br>ArrayData[x][y] = Rand.nextInt(100);}<br>}<br>}<br>}<br>} |       |



| Question | Answer   | Marks    |
|----------|--|----------|
| 2(b)(i)  | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• 1st outer loop (dimension 1)</li> <li>• 2nd loop (dimension 2)</li> <li>• inner for loop for all second dimension</li> </ul><br><ul style="list-style-type: none"> <li>• Selection statement ...</li> <li>• ...swapping the numbers correctly</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Dim TempNumber As Integer Dim ArrayLength As Integer = 10 For X = 0 To ArrayLength - 1   For Y = 0 To ArrayLength - 2     For Z = 0 To ArrayLength - Y - 2       if ArrayData(X, Z) &gt; ArrayData(X, Z + 1) then         TempNumber = ArrayData(X, Z)         ArrayData(X, Z) = ArrayData(X, Z+1)         ArrayData(X, Z + 1) = TempNumber       end if     Next Z   Next Y Next X</pre> <p><b>Python</b></p> <pre>ArrayLength = 10 for X in range(0, ArrayLength):   for Y in range(0, ArrayLength-1):     for Z in range(0, ArrayLength - Y - 1):       if(ArrayData[X][Z] &gt; ArrayData[X][Z+1]):         TempNumber = ArrayData[X][Z]         ArrayData[X][Z] = ArrayData[X][Z+1]         ArrayData[X][Z+1] = TempNumber</pre> <p>Accept for MP5:<br/> ArrayData[X][Z], ArrayData[X][Z+1] = ArrayData[X][Z+1], ArrayData[X][Z]</p> | <b>5</b> |



| Question | Answer  | Marks |
|----------|---|-------|
| 2(b)(i)  | <b>Java</b><br>Integer ArrayLength = 10;<br>for(int X = 0; X < ArrayLength; X++){<br>for(int Y = 0; Y < ArrayLength; Y++){<br>for(int Z = 0; Z < ArrayLength - Y - 1; Z++){<br>if(ArrayData[X][Z] > ArrayData[X][Z + 1]){<br>TempNumber = ArrayData[X][Z];<br>ArrayData[X][Z] = ArrayData[X][Z+1];<br>ArrayData[X][Z + 1] = TempNumber;<br>}<br>}<br>}<br>} |       |



| Question | Answer  | Marks |
|----------|---|-------|
| 2(b)(ii) | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• procedure header (and end where appropriate)</li> <li>• Outputting all 10 × 10 values with each 2nd dimension on a complete line</li> <li>• Calling procedure before and after bubble sort</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre> Sub Main()     Dim random As New Random     Dim ArrayData(10, 10) As Integer     For x = 0 To 9         For y = 0 To 9             ArrayData(x, y) = random.Next(1, 100)         Next     Next     Console.WriteLine("before")     printarray(ArrayData)      Dim TempNumber As Integer     Dim ArrayLength As Integer = 10     For X = 0 To ArrayLength - 1         For Y = 0 To ArrayLength - 2             For Z = 0 To ArrayLength - Y - 2                 if ArrayData(X, Z) &gt; ArrayData(X, Z + 1) then                     TempNumber = ArrayData(X, Z)                     ArrayData(X, Z) = ArrayData(X, Z+1)                     ArrayData(X, Z + 1) = TempNumber                 end if             Next Z         Next Y     Next X      Console.WriteLine("after")     printarray(ArrayData)     Console.ReadLine() End Sub </pre> | 3     |



| Question | Answer  | Marks |
|----------|---|-------|
| 2(b)(ii) | <pre> Sub Printarray(ByRef ArrayData(,) As Integer)   For x = 0 To 9     For y = 0 To 9       Console.Write(ArrayData(x, y) &amp; " ")     Next     Console.WriteLine()   Next End Sub  <b>Python</b> import random  def Printarray(ArrayData):     for x in range(0, 10):         for y in range(0, 10):             print(ArrayData[x][y], " ", end='')         print("") #main ArrayData= [[0]*10 for i in range(10)] #integer for x in range(0, 10):     for y in range(0,10):         ArrayData[x][y] = random.randint(1, 100) print("Before") printarray(ArrayData) ArrayLength = 10 for X in range(0, ArrayLength):     for Y in range(0, ArrayLength):         for Z in range(0, ArrayLength - Y - 1):             if(ArrayData[X][Z] &gt; ArrayData[X][Z+1]):                 TempNumber = ArrayData[X][Z]                 ArrayData[X][Z] = ArrayData[X][Z+1]                 ArrayData[X][Z+1] = TempNumber  print("After") Printarray(ArrayData) </pre> |       |



| Question | Answer   | Marks |
|----------|--|-------|
| 2(b)(ii) | <pre> <b>Java</b> import java.util.Scanner; import java.util.Random;  class Question2{      public static Integer[][] ArrayData;      public static void printArray(Integer[][] theArrayData){         for(int x = 0; x &lt; 10; x++){             for(int y = 0; y &lt; 10; y++){                 System.out.printf(theArrayData[x][y] + " ");             }             System.out.println();         }     }      public static void main(String args[]){         Random rand = new Random();         ArrayData = new Integer[10][10];         for(int x=0; x &lt; 10; x++){             for(int y = 0; y &lt; 10; y++){                 ArrayData[x][y] = rand.nextInt(100);             }         }         Integer TempNumber = 0;          System.out.println("Before");         printArray(ArrayData);         Integer ArrayLength = 10;         for(int X = 0; X &lt; ArrayLength; X++){             for(int Y = 0; Y &lt; ArrayLength; Y++){                 for(int Z = 0; Z &lt; ArrayLength - Y - 1; Z++){                     if(ArrayData[X][Z] &gt; ArrayData[X][Z + 1]){ </pre> |       |



| Question | Answer  | Marks |
|----------|---|-------|
| 2(b)(ii) | <pre>TempNumber = ArrayData[X][Z]; ArrayData[X][Z] = ArrayData[X][Z+1]; ArrayData[X][Z + 1] = TempNumber;     }   } } System.out.println("After"); PrintArray(ArrayData); } }</pre> |       |


| Question  | Answer  | Marks |
|-----------|---|-------|
| 2(b)(iii) | <p>1 mark for output showing array unsorted and then sorted on 1 of the dimensions<br/>e.g.</p> <pre> before 87 37 36 72 85 58 82 34 6 90 71 44 24 26 47 80 22 28 68 70 45 52 95 31 18 91 91 73 19 10 25 52 47 4 78 70 46 62 74 49 38 9 95 80 16 81 99 69 78 72 81 98 95 43 96 35 42 6 54 93 67 9 16 95 65 20 97 20 48 76 35 59 54 20 33 36 56 16 15 36 91 22 95 46 29 27 57 4 8 43 29 41 9 80 81 29 93 1 15 96 after 6 34 36 37 58 72 82 85 87 90 22 24 26 28 44 47 68 70 71 80 10 18 19 31 45 52 73 91 91 95 4 25 46 47 49 52 62 70 74 78 9 16 38 69 72 78 80 81 95 99 6 35 42 43 54 81 93 95 96 98 9 16 20 20 48 65 67 76 95 97 15 16 20 33 35 36 36 54 56 59 4 8 22 27 29 43 46 57 91 95 1 9 15 29 29 41 80 81 93 96 </pre> | 1     |

| Question | Answer   | Marks |
|----------|--|-------|
| 2(c)(i)  | <p>1 mark for each completed statement (6)<br/>1 mark per mark point</p> <ul style="list-style-type: none"> <li>function declaration taking appropriate parameters <b>and</b> recursive calls</li> <li>remainder of the function is accurate including appropriate DIV operator.</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Function BinarySearch(ByVal SearchArray(,) As Integer, Lower As Integer, Upper As Integer, SearchValue As Integer)     Dim Mid As Integer     If Upper &gt;= 0 Then         Mid = (Lower + (Upper - 1)) \ 2         If SearchArray(0, Mid) = <b>SearchValue</b> Then             Return <b>Mid</b>         ElseIf SearchArray(0, Mid) &gt; SearchValue Then             Return BinarySearch(SearchArray, <b>Lower</b>, Mid - 1, SearchValue)         Else             Return BinarySearch(SearchArray, Mid + 1, <b>Upper</b>, SearchValue)         End If     End If     Return <b>-1</b> End Function</pre> <p><b>Python</b></p> <pre>def BinarySearch(SearchArray, Lower, Upper, SearchValue):     if Upper &gt;= 0:         Mid = int((Lower + (Upper - 1)) / 2)         If SearchArray[0][Mid] == <b>SearchValue</b>:             return <b>Mid</b>         elif SearchArray[0][Mid] &gt; SearchValue:             return BinarySearch(SearchArray, <b>Lower</b>, Mid-1, SearchValue)         else:             return BinarySearch(SearchArray, Mid+1, <b>Upper</b>, SearchValue)     return <b>-1</b></pre> | 8     |



| Question | Answer   | Marks |
|----------|--|-------|
| 2(c)(i)  | <b>Java</b><br><pre>public static Integer BinarySearch(Integer[][] SearchArray, Integer Lower, Integer Upper, Integer SearchValue){     Integer Mid = 0;     If Upper &gt;= 0 {         Mid = (Lower + (Upper - 1)) / 2;         If SearchArray[0][Mid] == <b>SearchValue</b> ){             return <b>Mid</b>;         }else if SearchArray[0][Mid] &gt; SearchValue {             return BinarySearch(SearchArray, <b>Lower</b>, Mid-1, SearchValue);         }else{             return BinarySearch(SearchArray, Mid+1, <b>Upper</b>, SearchValue);         }     } return <b>-1</b>; }</pre> |       |



| Question | Answer   | Marks |
|----------|--|-------|
| 2(c)(ii) | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• screenshot outputting the index when Number is found</li> <li>• screenshot outputting –1 with a Number not found</li> </ul> <p>e.g.</p>  <pre> before 85 61 28 29 47 62 7 13 10 5 85 68 13 63 43 70 12 92 33 34 30 55 62 7 4 31 45 18 63 49 83 58 68 76 9 23 28 77 21 68 25 8 99 14 51 32 66 85 49 94 72 31 71 83 25 31 99 22 26 16 18 86 48 60 21 28 9 37 55 21 43 35 71 66 10 38 56 49 56 71 64 59 68 69 76 53 86 77 51 78 24 7 52 6 49 36 68 39 92 23 after 5 7 10 11 28 29 47 61 62 85 12 13 33 34 43 63 68 70 85 92 4 7 18 31 38 45 49 55 62 63 9 21 23 28 58 60 66 76 77 83 8 14 25 32 49 51 66 85 94 99 16 22 25 26 31 31 71 72 83 99 9 18 21 21 28 37 48 55 60 68 16 38 35 43 49 56 56 66 71 71 51 53 59 64 68 69 76 77 78 86 6 7 23 24 36 39 49 52 60 92 10 2 10 1 </pre> | 2     |



| Question | Answer   | Marks    |
|----------|--|----------|
| 3(a)     | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• Card class declaration (and end where appropriate)</li> <li>• Both attributes (Number and Colour) declared with suitable data types ...</li> <li>• ...as private</li> <li>• correct constructor header (and end where appropriate) with <b>two</b> parameters ...</li> <li>• ...both <b>parameters</b> assigned to the attributes</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Class Card   Private Number As Integer   Private Colour As String   Sub New(Numberp, Colourp)     Number = Numberp     Colour = Colourp   End Sub End Class</pre> <p><b>Python</b></p> <pre>class Card:   #Number as Integer   #Colour as string   def __init__(self, Numberp, Colourp):     self.__Number = Numberp     self.__Colour = Colourp</pre> | <b>5</b> |



| Question | Answer   | Marks |
|----------|--|-------|
| 3(a)     | <b>Java</b><br><pre>import java.util.Scanner; import java.io.*; class Card{     private Integer Number;     private String Colour;     public Card(Integer pNumber, String pColour){         Number = pNumber;         Colour = pColour;     }     public static void main(String args[]){     } }</pre> |       |



| Question | Answer   | Marks    |
|----------|--|----------|
| 3(b)     | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• 1 get method header (and close where appropriate) with no parameter ...</li> <li>• ... returning attribute</li> <li>• 2nd correct get method</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Function GetNumber()     Return Number End Function Function GetColour()     Return Colour End Function</pre> <p><b>Python</b></p> <pre>def GetNumber(self):     return self.__Number def GetColour(self):     return self.__Colour</pre> <p><b>Java</b></p> <pre>public Integer GetNumber(){     return Number; }  public String GetColour(){     return Colour; }</pre> | <b>3</b> |



| Question | Answer  | Marks    |
|----------|---|----------|
| 3(c)     | <p>1 mark per mark point to max 7</p> <ul style="list-style-type: none"> <li>• Declaration of array with 30 elements of type Card</li> <li>• Opening the text file CardValues.txt for read</li> <li>• Looping until EOF/30 times</li> <li>• Reading in <b>all</b> sets of <b>2</b> lines (number then colour) ...</li> <li>• ...creating object of type Card...</li> <li>• ...with number and colour read in from file ...</li> <li>• ...storing in next array element</li> <li>• Try and catch for file handling...s</li> <li>• ...with appropriate outputs</li> <li>• Closing the file in a <b>suitable</b> place</li> </ul> <p>Example program code:<br/><b>VB.NET</b></p> <pre>Sub Main()     Dim CardArray(0 To 29) As Card     Dim NumberRead As Integer     Dim ColourRead As String      Try         Dim Filename As String = "CardValues.txt"         Dim FileReader As New System.IO.StreamReader(filename)         For x = 0 To 29             NumberRead = FileReader.ReadLine()             ColourRead = FileReader.ReadLine()             CardArray(x) = New Card(NumberRead, ColourRead)         Next         FileReader.close()         Catch ex As Exception</pre> | <b>7</b> |



| Question | Answer  | Marks |
|----------|---|-------|
| 3(c)     | <pre>         Console.WriteLine("Invalid file")         End Try     End Sub  <b>Python</b> CardArray = [0,0] #integer  try: Filename = "CardValues.txt" File = open(Filename,'r') for x in range(0,30): NumberRead = int(File.readline()) ColourRead = File.readline() CardArray[x] = Card(NumberRead, ColourRead)  File.close except IOError: print("Could not find file")  <b>Java</b> public static void main(String args[]){     Card[] CardArray = new Card[30];     Integer NumberRead;     String ColourRead;     String FileName = "CardValues.txt";     try{         FileReader F = new FileReader(FileName);         BufferedReader Reader = new BufferedReader(f);         for(Integer x = 0; x &lt; 30; x++){             NumberRead = Integer.parseInt(Reader.readLine());             ColourRead = Reader.readLine();             CardArray[x] = new Card(NumberRead, ColourRead);         }         Reader.close();     } } </pre> |       |



| Question | Answer   | Marks |
|----------|--|-------|
| 3(c)     | <pre>catch(FileNotFoundException ex){     System.out.println("No file found"); } catch(IOException ex){     System.out.println("No file found"); } }</pre> |       |



| Question | Answer  | Marks    |
|----------|---|----------|
| 3(d)     | <p>1 mark per mark point to max 6</p> <ul style="list-style-type: none"> <li>• Implementing a suitable way of storing which card have been selected</li> <li>• Function ChooseCard( ) header (and close) <b>and</b> returning an integer (index) in all cases</li> <li>• Reading in array index from the user ...</li> <li>• ...with suitable validation <b>looping</b> until it is between 1 and 30 (inclusive)</li> <li>• Converting input to array index (e.g. –1 each time)</li> <li>• Check if the input is already selected...</li> <li>• ... if it is selected, loop until index input is not already selected</li> <li>• ... returning index of available card selected</li> <li>• Stores the valid Card chosen as taken (using any suitable method)</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Dim NumbersChosen(0 To 29) As Boolean Sub Main()     For x = 0 To 29         NumbersChosen(x) = False     Next     ... End Sub Function chooseCard()     Dim CardSelected As Integer     Dim flagContinue As Boolean = True     While flagContinue = True         Console.WriteLine("Select a Card from 1 to 30")         CardSelected = Console.ReadLine()         If CardSelected &lt; 1 Or CardSelected &gt; 30 Then             Console.WriteLine("Number must be between ")         End If     End While End Function</pre> | <b>6</b> |



| Question | Answer   | Marks |
|----------|--|-------|
| 3(d)     | <pre> ElseIf NumbersChosen(CardSelected - 1) = True Then   Console.WriteLine("Already taken") Else   Console.WriteLine("valid")   flagContinue = False End If End While NumbersChosen(CardSelected - 1) = True Return CardSelected - 1 End Function  <b>Python</b> global NumbersChosen ... def chooseCard ():   global NumbersChosen   flagContinue = True   while flagContinue == true:     CardSelected = int(input("Select a Card from 1 to 30"))     if CardSelected &lt; 1 or CardSelected &gt; 30:       print("Number must be between 1 and 30")     elif NumbersChosen(CardSelected - 1) == True:       print("Already taken")     else:       print("Valid")       flagContinue = False   NumbersChosen[CardSelected-1] = True   return CardSelected-1 ... #main ... NumbersChosen = [False for i in range(30)] </pre> |       |





| Question | Answer   | Marks |
|----------|--|-------|
| 3(d)     | <pre><b>Java</b> public static Boolean[] NumbersChosen = new Boolean[30];  public Integer chooseCard (){     Boolean flagContinue = true;     Integer CardSelected = -1;     while(flagContinue){         System.out.println("Select a Card from 1 to 30");         Scanner scanner = new Scanner(System.in);         CardSelected = Integer.parseInt(scanner.nextLine());         if(CardSelected &lt; 1    CardSelected &gt; 30){             System.out.println("Number must be between 1 and 30");          }else if(NumbersChosen[CardSelected - 1]){             System.out.println("Already taken");         }else{             System.out.println("Valid");             flagContinue = false;         }     }     NumbersChosen[CardSelected - 1] = true;     return CardSelected - 1; }</pre> |       |



| Question | Answer  | Marks    |
|----------|---|----------|
| 3(e)(i)  | <p>1 mark per mark point</p> <ul style="list-style-type: none"> <li>• declaring array <code>Player1</code> of type <code>Card</code></li> <li>• calling the function <code>ChooseCard()</code> <b>four</b> times</li> <li>• storing the <b>card</b>, that is in the index returned, in the array <code>Player1</code></li> </ul><br><ul style="list-style-type: none"> <li>• outputting all <b>four</b> numbers <b>and</b> colours in <code>Player1</code> ...</li> <li>• .... using the get methodss</li> </ul> <p>Example program code:</p> <p><b>VB.NET</b></p> <pre>Dim Player1(0 To 3) As Card For x = 0 To 3     Player1(x) = CardArray(ChooseCard(NumbersChosen)) Next for x = 0 to 3     console.writeline(Player1(x).GetColour)     console.writeline(Player1(x).GetNumber) next x</pre> <p><b>Python</b></p> <pre>Player1 = [] #of type Card for x in range(0, 4):     ReturnNumber = ChooseCard ()     Player1.append(CardArray[ReturnNumber]) for x in range(0, 4):     print(Player1[x].GetColour())     print(Player1[x].GetNumber())</pre> | <b>5</b> |



| Question | Answer   | Marks    |
|----------|--|----------|
| 3(e)(i)  | <p><b>Java</b></p> <pre> Card[] Player1 = new Card[5]; for(Integer x = 0; x &lt; 5; x++){     Player1[x] = CardArray[ChooseCard()]; } for(Integer x = 0; x &lt; 5; x++){     System.out.println(Player1[x].GetColour());     System.out.println(Player1[x].GetNumber()); } </pre>  |          |
| 3(e)(ii) | <p>1 mark for both tests<br/> Test 1: inputting 1, 5, 9, 10. Outputting: 1 red 9 green 9 orange 10 red<br/> Test 2: inputting 2 2 3 4 4 5. Outputting: 2 already taken. Then 5 black 2 while 4 red 9 green<br/> Test 1 e.g.</p> <pre> Select a card from 1 to 301 Valid Select a card from 1 to 305 Valid Select a card from 1 to 309 Valid Select a card from 1 to 3010 Valid 1 red  9 green  9 orange  10 red </pre> | <b>1</b> |

**PUBLISHED**

| Question | Answer   | Marks |
|----------|--|-------|
| 3(e)(ii) | <p>Test 2 e.g.</p> <p>Select a card from 1 to 302<br/>Valid</p> <p>Select a card from 1 to 302<br/>Already taken</p> <p>Select a card from 1 to 303<br/>Valid</p> <p>Select a card from 1 to 304<br/>Valid</p> <p>Select a card from 1 to 304<br/>Already taken</p> <p>Select a card from 1 to 305<br/>Valid</p> <p>5<br/>black</p> <p>2<br/>white</p> <p>4<br/>red</p> <p>9<br/>green</p> |       |

