

10	One mark per mark point (Max 2) MP1 Exception handling is a process that responds to unwanted / unexpected events when a program runs MP2 ... to prevent the program / computer from stopping unexpectedly One mark for example (Max 1) MP3 Programming errors MP4 User errors MP5 Hardware failure // losing connection to a device such as a printer
----	---

10(a)	One mark per mark point (Max 2) MP1 Use an exception handling routine (to respond to unwanted / unexpected events when the program is running) MP2 Use of try ... except / catch // Generate some form of error message
10(b)	One mark per mark point (Max 2) MP1 Coding errors MP2 User errors MP3 Hardware failure // Losing connection to a device e.g. a printer

12(a)	One mark per mark point (Max 3) MP1 An exception is an unplanned / unexpected event that occurs when a program is running MP2 ... due to an error in programming/logic that wasn't detected during program construction/compilation MP3 The effect of an exception can be that the program halts unexpectedly.
12(b)	One mark per mark point (Max 2) MP1 One mark for an example of an exception (see list in guidance) MP2 One mark for the reason for the given exception Example answer Division by zero The processor will not be able to evaluate this answer because a number divided by zero is infinity, so the program will crash.

2(a)	1 mark each to max 7 - Function header (and end) - Prompt to enter filename and reading input - Opening the file (to read) and closing the file in an appropriate place - Looping until EOF reading each line in the file (removing line break and) inserting in array - Returning populated array - Exception handling try catch with appropriate output
------	---

2(b)	1 mark each - Procedure header (and end) taking (1D array) <code>dataArray</code> (of strings) as a parameter - Declaration/use of 6 1D arrays (equivalent), one for each colour - Looping through each line in parameter <code>dataArray</code> splitting by comma - Comparing 2nd value/colour to each colour to select array storing 1st value/integer in correct array
------	--

2(c)	1 mark each - Procedure header taking (1D array and filename as parameters, opening file to append and closing file (in appropriate place) - Looping through each item in array parameter writing to the file ... with new line break between each line - Using exception handling try and catch with suitable output
------	--

Example program code:

```
Python
def StoreData(DataToStore, FileName):
    try:
        File = open(FileName, "a+")
        for Item in DataToStore:
            File.write(Item)
            File.write("\n")
        File.close()
    except:
        print("Cannot create or write to file")
```

```
VB.NET
Sub StoreData(DataToStore(), FileName)
    Dim FileWriter As IO.StreamWriter = New IO.StreamWriter(FileName, False)
    Dim x As Integer = 0
    Try
        While DataToStore(x) IsNot Nothing
            FileWriter.WriteLine(DataToStore(x))
            x = x + 1
        End While
        FileWriter.Close()
    Catch ex As Exception
        Console.WriteLine("Cannot open or write to file")
    End Try
End Sub
```

2(d)	1 mark each - Calling StoreData with one array and filename - Calling StoreData with remaining 5 arrays and filename
Example program code: Python StoreData(Red, "Red.txt") StoreData(Green, "Green.txt") StoreData(Blue, "Blue.txt") StoreData(Orange, "Orange.txt") StoreData(Yellow, "Yellow.txt") StoreData(Pink, "Pink.txt") VB.NET StoreData(Red, "Red.txt") StoreData(Green, "Green.txt") StoreData(Blue, "Blue.txt") StoreData(Orange, "Orange.txt") StoreData(Yellow, "Yellow.txt") StoreData(Pink, "Pink.txt") Java StoreData(Red, "Red.txt"); StoreData(Green, "Green.txt"); StoreData(Blue, "Blue.txt"); StoreData(Orange, "Orange.txt"); StoreData(Yellow, "Yellow.txt"); StoreData(Pink, "Pink.txt");	

2(e)(i)	1 mark each - Calling ReadData() and storing/using return value - Calling SplitData() with returned array as a parameter
Example program code: Python DataFromFile = ReadData() SplitData(DataFromFile) VB.NET Sub Main(args As String()) Dim DataFromFile(,) As String = ReadData() SplitData(DataFromFile) End Sub Java public static void main(String args[]){ String[][] DataFromFile = ReadData(); SplitData(DataFromFile); }	
2(e)(ii)	1 mark screenshots showing - Prompt and input of filename TheData.txt - Screenshot of data in red file. Filename must be shown in same screenshot as data

3(a)	1 mark each - HighScores created as 2D array, (of strings) with (min) 7 × 3 elements (local to main) all elements initialised to empty string ("") e.g. Python HighScores = [] #String, 7 x 3 HighScores = [['' for x in range(3)] for y in range(7)] VB.NET Dim HighScores(7, 3) As String For(X = 0 to 7) For(Y = 0 to 3) HighScores(X, Y) = "" Next Y Next X Java String[][] HighScores = new String[7][3]; for(Int X = 0; X < 7; X++){ for(Int Y = 0; Y < 3; Y++){ HighScores[X][Y] = ""; } }
3(b)	1 mark each - Function header (and end where appropriate) that returns populated array - Opening text file to read and closing the file in an appropriate place - Looping through 7 players/to EOF/21 times reading in each group of 3 data items and storing each in separate element in 2D array for each player - Exception handling with all file handling within the try, appropriate catch and an output
3(b)	e.g. Python def ReadData(): Temp = [] HighScores = [] try: File = open("HighScoreTable.txt") Temp = File.read().split("\n") File.close() except: print("No file found") NumberRecords = len(Temp)-1 Counter = 0 while Counter < NumberRecords: HighScores.append([Temp[Counter], Temp[Counter+1], Temp[Counter+2]]) Counter = Counter + 3 return HighScores VB.NET Function ReadData() Dim TextFile As String = "HighScoreTable.txt" Dim HighScores(7, 3) As String Try Dim FileReader As New System.IO.StreamReader(TextFile) Dim Counter As Integer = 0

3(b)	<pre> While Counter < 8 HighScores(Counter, 0) = FileReader.ReadLine() HighScores(Counter, 1) = FileReader.ReadLine() HighScores(Counter, 2) = FileReader.ReadLine() Counter = Counter + 1 End While FileReader.Close() Catch ex As Exception Console.WriteLine("No file found") End Try Return HighScores End Function Java public static String[][] ReadData(){ String TextFile = "HighScoreTable.txt"; String[][] HighScores = new String[7][3]; try{ FileReader f = new FileReader(TextFile); BufferedReader Reader = new BufferedReader(f); for(Integer X = 0; X < 7; X++){ try{ </pre>
------	--

3(b)	<pre> HighScores[X][0] = Reader.readLine(); HighScores[X][1] = Reader.readLine(); HighScores[X][2] = Reader.readLine(); }catch(IOException ex){} } try{ Reader.close(); }catch(IOException ex){} return HighScores; }catch(FileNotFoundException e){ System.out.println("File not found"); } return HighScores; } </pre>
3(c)	1 mark each - Procedure (header and end) taking (min) 1 parameter (2D array), looping through each of the first dimension in array outputting all data in correct format e.g. Python <pre> def OutputHighScores(HighScores): for x in range(0, len(HighScores)): print(HighScores[x][0], "reached level", HighScores[x][1], "with a score of", HighScores[x][2]) </pre> VB.NET <pre> Sub OutputHighScores(HighScores()) For x = 0 To 6 Console.WriteLine(HighScores(x, 0) & " reached level " & HighScores(x, 1) & " with a score of " & HighScores(x, 2)) Next End Sub </pre>

3(c)	Java <pre> public static void OutputHighScores(String[][] HighScores){ for(Integer x = 0; x < 7; x++){ System.out.println(HighScores[x][0] + " reached level " + HighScores[x][1] + " with a score of " + HighScores[x][2]); } } </pre>
3(d)	1 mark each - Function header (and end taking array as parameter) returning sorted array. - Comparing the levels and swapping all dimensions when in incorrect order - Comparing scores when levels are the same and swapping all dimensions when in incorrect order - Correct loops and comparisons to put data in correct order e.g. Python <pre> def SortScores(HighScores): Counter = 0 ArrayLength = len(HighScores) for x in range(ArrayLength-1): for y in range(0, ArrayLength-x-1): if int(HighScores[y][1]) < int(HighScores[y + 1][1]): HighScores[y], HighScores[y + 1] = HighScores[y + 1], HighScores[y] elif int(HighScores[y][1]) == int(HighScores[y+1][1]): if int(HighScores[y][2]) < int(HighScores[y+1][2]): HighScores[y], HighScores[y + 1] = HighScores[y + 1], HighScores[y] return HighScores </pre> VB.NET <pre> Function SortScores(HighScores) Dim ArrayLength As Integer = 6 Dim Temp1 As String </pre>

3(d)	<pre> Dim Temp2 As String Dim Temp3 As String For x = 0 To ArrayLength - 1 For y = 0 To ArrayLength - x - 1 If Integer.Parse(HighScores(y, 1)) < Integer.Parse(HighScores(y + 1, 1)) Then Temp1 = HighScores(y, 0) Temp2 = HighScores(y, 1) Temp3 = HighScores(y, 2) HighScores(y, 0) = HighScores(y + 1, 0) HighScores(y, 1) = HighScores(y + 1, 1) HighScores(y, 2) = HighScores(y + 1, 2) HighScores(y + 1, 0) = Temp1 HighScores(y + 1, 1) = Temp2 HighScores(y + 1, 2) = Temp3 ElseIf Integer.Parse(HighScores(y, 1)) = Integer.Parse(HighScores(y + 1, 1)) Then If Int(HighScores(y, 2)) < Int(HighScores(y + 1, 2)) Then Temp1 = HighScores(y, 0) Temp2 = HighScores(y, 1) Temp3 = HighScores(y, 2) HighScores(y, 0) = HighScores(y + 1, 0) HighScores(y, 1) = HighScores(y + 1, 1) HighScores(y, 2) = HighScores(y + 1, 2) HighScores(y + 1, 0) = Temp1 HighScores(y + 1, 1) = Temp2 HighScores(y + 1, 2) = Temp3 End If End If Next y Next x Return HighScores End Function </pre> Java <pre> public static String[][] SortScores(String[][] HighScores){ </pre>
------	--

3(d)	<pre> Integer ArrayLength = 6; String Temp1; String Temp2; String Temp3; for(Integer x = 0; x < ArrayLength; x++){ for(Integer y = 0; y < ArrayLength - x; y++){ if(Integer.parseInt(HighScores[y][1]) < Integer.parseInt(HighScores[y+1][1])){ Temp1 = HighScores[y][0]; Temp2 = HighScores[y][1]; Temp3 = HighScores[y][2]; HighScores[y][0] = HighScores[y+1][0]; HighScores[y][1] = HighScores[y+1][1]; HighScores[y][2] = HighScores[y+1][2]; HighScores[y+1][0] = Temp1; HighScores[y+1][1] = Temp2; HighScores[y+1][2] = Temp3; }else if(Integer.parseInt(HighScores[y][1]) == Integer.parseInt(HighScores[y+1][1])){ if(Integer.parseInt(HighScores[y][2]) < Integer.parseInt(HighScores[y+1][2])){ Temp1 = HighScores[y][0]; Temp2 = HighScores[y][1]; Temp3 = HighScores[y][2]; HighScores[y][0] = HighScores[y+1][0]; HighScores[y][1] = HighScores[y+1][1]; HighScores[y][2] = HighScores[y+1][2]; HighScores[y+1][0] = Temp1; HighScores[y+1][1] = Temp2; HighScores[y+1][2] = Temp3; } } } } </pre>
------	--

3(d)	<pre> } return HighScores; } </pre>
------	---

3(e)(i)	1 mark each Code statements in order: HighScores = ReadData() HighScores = SortScores(HighScores) // HighScores = SortScores() Code statements in order: OUTPUT "Before" OutputHighScores(HighScores) unsorted OUTPUT "After" OutputHighScores(HighScores) sorted e.g. Python HighScores = [] HighScores = ReadData() print("Before") OutputHighScores(HighScores) HighScores = SortScores(HighScores) print("After") OutputHighScores(HighScores) VB.NET Sub Main(args As String()) Dim HighScores(7, 3) As String HighScores = ReadData() Console.WriteLine("Before") OutputHighScores(HighScores) HighScores = SortScores(HighScores) Console.WriteLine("After") OutputHighScores(HighScores) End Sub
---------	---

3(e)(i)	Java <pre> public static void main(String args[]){ String[][] HighScores = new String[7][3]; HighScores = ReadData(); System.out.println("Before"); OutputHighScores(HighScores); HighScores = SortScores(HighScores); System.out.println("After"); OutputHighScores(HighScores); } </pre>
3(e)(ii)	Output showing 'Before' and players and scores in correct format before sorting Output showing 'After' players and scores in correct order and format after sorting e.g. <pre> Before GHEH got to level 3 with a score of 10 RWQW got to level 4 with a score of 20 MMND got to level 4 with a score of 18 RFOO got to level 5 with a score of 20 XXHD got to level 3 with a score of 19 QWSD got to level 3 with a score of 15 JGHF got to level 5 with a score of 22 After JGHF got to level 5 with a score of 22 RFOO got to level 5 with a score of 20 RWQW got to level 4 with a score of 20 MMND got to level 4 with a score of 18 XXHD got to level 3 with a score of 19 QWSD got to level 3 with a score of 15 GHEH got to level 3 with a score of 10 </pre>