

Cambridge International AS & A Level

COMPUTER SCIENCE**9618/41**

Paper 4 Practical

May/June 2026**MARK SCHEME**Maximum Mark: 75

Published

This mark scheme contains the instructions and marking guidance that our examiners used to mark candidate responses for this component.

Before they start marking, examiners complete a standardisation process. They review a range of candidate responses and discuss the mark scheme in detail to agree which answers can and cannot be accepted. Examiners award marks where it is clear that candidate responses have met the requirements of the mark scheme. These requirements may vary between different components or different versions of the same component, depending on the exact requirements of the question and the candidate responses seen during the standardisation process.

Sometimes, our mark schemes may allow marks to be awarded to responses which contain elements that are factually incorrect or for content not covered by the syllabus. We do this in response to the demands of a specific question to support candidates and make sure that our assessments are fair. However, these allowances should not be used as a guide for teaching and learning.

We cannot discuss the mark scheme with schools, and we recommend that schools read the mark scheme together with the assessment material and the Principal Examiner Report for Teachers.

This document consists of **41** printed pages.

General marking principles

All examiners must apply these marking principles when marking candidate responses.

1	Examiners must award marks for each response in line with: - the specific content defined in the mark scheme - the specific skills defined in the mark scheme or in the level descriptions - the standard of response required as exemplified by the standardisation scripts.
2	Examiners must award whole marks (not half marks, or other fractions).
3	Examiners must award marks positively . This means that: - marks are not deducted for errors or omissions - marks are awarded for correct/valid answers as defined in the mark scheme and also for other valid answers which go beyond the scope of the syllabus and mark scheme referring to your team leader as appropriate - the quality of spelling, punctuation and grammar are judged only when the mark scheme indicates that these features are specifically assessed in the question. However, the meaning of all responses must be unambiguous.
4	Examiners must consistently apply the requirements of the mark scheme and any rules for what to do when candidates have not followed instructions correctly.
5	Examiners must use the full range of marks defined in the mark scheme, unless limited by the quality of the candidate responses seen.
6	Examiners must award marks according to the mark scheme and must not award marks with grade thresholds or grade descriptions in mind.

Annotations guidance for centres

Examiners use a system of annotations as a shorthand for communicating their marking decisions to one another. Examiners are trained during the standardisation process on how and when to use annotations. The purpose of annotations is to inform the standardisation and monitoring processes and guide the supervising examiners when they are checking the work of examiners within their team. The meaning of annotations and how they are used is specific to each component and is understood by all examiners who mark the component.

We publish annotations in our mark schemes to help centres understand the annotations they may see on copies of scripts. Note that there may not be a direct correlation between the number of annotations on a script and the mark awarded. Similarly, the use of an annotation may not be an indication of the quality of the response.

The annotations listed below were available to examiners marking this component in this series.

Annotations

Annotation	Meaning
	Benefit of the doubt
	To indicate where a key word/phrase/code is missing
	Incorrect
	Follow through
	Indicate a point in an answer
Highlighted text	To draw attention to a particular aspect or to indicate where parts of an answer have been combined
	Ignore
	Not answered question
	No benefit of doubt given
	No examples or not enough

Annotation	Meaning
	Not relevant or used to separate parts of an answer
Off-page comment	Allows comments to be entered at the bottom of the RM marking window and then displayed when the associated question item is navigated to.
	Repetition
	Indicates that work or a page has been seen including blank answer spaces and blank pages.
	Correct
	Too vague

PUBLISHED

- **Bold** in mark scheme means that idea is required.
- / in mark scheme means alternative.
- // in mark scheme means alternative solution that gains the same mark point.
- ... at the end of one mark point without a ... at the start of the next just means the sentence follows on. There is no dependency.
- ... at the end of one mark point and ... at the start of the next, this means the second cannot be awarded without the first.
- () means what is in the brackets is not required, or it is not required in some languages but may be required in others.

Question	Answer	Marks	Guidance
1(a)	1 mark each • (Global) array of 10 integers • (Global) head pointer and tail pointer both initialised to -1 • Queue values are initialised to -1	3	

Question	Answer	Marks	Guidance
	Example program code Python <pre>Queue = [] HeadPointer = -1 TailPointer = -1 for x in range(10): Queue.append(-1)</pre> VB.NET <pre>Dim Queue(9) As Integer Dim HeadPointer As Integer Dim TailPointer As Integer Sub Main(args As String()) HeadPointer = -1 TailPointer = -1 For x = 0 To 9 Queue(x) = -1 Next End Sub</pre> Java <pre>public static Integer[] Queue = new Integer[10]; public static Integer HeadPointer; public static Integer TailPointer; public static void main(String args[]){ HeadPointer = -1; TailPointer = -1; for(Integer x = 0; x < 10; x++){ Queue[x] = -1; } }</pre>		

PUBLISHED

Question	Answer	Marks	Guidance
1(b)	1 mark each • Enqueue header taking 1 (integer) parameter • Checking if the queue is full ... • ... returning False • (otherwise) Storing parameter in tail pointer +1 • Incrementing tail pointer • ... returning True • Dealing with first element and changing head pointer to be 0	7	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def Enqueue(Data): global Queue global TailPointer global HeadPointer if TailPointer < 9: TailPointer = TailPointer + 1 Queue[TailPointer] = Data if HeadPointer == -1: HeadPointer = 0 return True else: return False</pre> VB.NET <pre>Function Enqueue(DataValue As Integer) If TailPointer < 9 Then TailPointer = TailPointer + 1 Queue(TailPointer) = DataValue If HeadPointer = -1 Then HeadPointer = 0 End If Return True Else Return False End If End Function</pre>		

Question	Answer	Marks	Guidance
Java	<pre>public static Boolean Enqueue(Integer DataValue){ if (TailPointer < 9){ TailPointer++; Queue[TailPointer] = DataValue; if(HeadPointer == -1){ HeadPointer = 0; } return true; }else{ return false; } }</pre>		

PUBLISHED

Question	Answer	Marks	Guidance
1(c)(i)	1 mark each • Taking 11 (integer) values as input • ... calling <code>Enqueue()</code> with each input • Checking/storing return value ... • ... output message if False is returned • Output content of array in any form	5	

Question	Answer	Marks	Guidance
	Example program code Python <pre>for x in range(11): Returned = Enqueue(int(input("Enter a number "))) if not(Returned): print("Queue full") print(Queue)</pre> VB.NET <pre>Dim Returned As Boolean For x = 0 To 10 Console.WriteLine("Enter a number ") Returned = Enqueue(Console.ReadLine()) If Not Returned Then Console.WriteLine("Queue full") End If Next</pre> Java <pre>Scanner scanner = new Scanner(System.in); Boolean Returned; for(Integer x = 0; x < 12; x++){ System.out.println("Enter a number"); Returned = Enqueue(Integer.parseInt(scanner.nextLine())); if(! Returned){ System.out.println("Queue full"); } } for(Integer x = 0; x < 10; x++){ System.out.println(Queue[x]); }</pre>		

Question	Answer	Marks	Guidance
1(c)(ii)	1 mark for ss showing inputs and output	1	
e.g. <pre> Enter a number 1 Enter a number 2 Enter a number 3 Enter a number 4 Enter a number 5 Enter a number 6 Enter a number 7 Enter a number 8 Enter a number 9 Enter a number 10 Enter a number 11 Queue full [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] </pre>			

Question	Answer	Marks	Guidance
1(d)	1 mark each • Dequeue header taking no parameter • Checking if queue is empty (e.g. TailPointer == -1 or HeadPointer > TailPointer) ... • ... returning –1 • (otherwise) accessing data at head pointer ... • ... and returning • Incrementing HeadPointer (before any return)	6	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def Dequeue(): global Queue global HeadPointer global TailPointer if HeadPointer > -1 and HeadPointer <= TailPointer: ReturnValue = Queue[HeadPointer] HeadPointer = HeadPointer + 1 return ReturnValue else: return -1</pre> VB.NET <pre>Function Dequeue() Dim ReturnValue As Integer If HeadPointer > -1 And HeadPointer <= TailPointer Then ReturnValue = Queue(HeadPointer) HeadPointer = HeadPointer + 1 Return ReturnValue Else Return -1 End If End Function</pre> Java <pre>public static Integer Dequeue(){ if (HeadPointer > -1 && HeadPointer <= TailPointer){ Integer ReturnValue = Queue[HeadPointer]; HeadPointer++; return ReturnValue; }else{ return -1; } }</pre>		

Question	Answer	Marks	Guidance
1(e)(i)	1 mark each • Taking 6 integers as input and storing in the queue • Calling <code>Dequeue()</code> 4 times and storing/using return value • Outputs each removed element from the queue	3	
Example program code Python <pre>for x in range(6): returnValue = Enqueue(int(input("Enter a number "))) for x in range(4): returnValue = Dequeue() print(returnValue)</pre> VB.NET <pre>For x = 0 To 5 Console.WriteLine("Enter a number ") Returned = Enqueue(Console.ReadLine()) Next Dim ReturnValue As Integer = -1 For x = 0 To 3 ReturnValue = Dequeue() Console.WriteLine(ReturnValue) Next</pre> Java <pre>for(Integer x = 0; x < 6; x++){ System.out.println("Enter a number"); Returned = Enqueue(Integer.parseInt(scanner.nextLine())); } Integer ReturnValue = -1; for(Integer x = 0; x < 4; x++){ ReturnValue = Dequeue(); System.out.println(ReturnValue); }</pre>			

Question	Answer	Marks	Guidance
1(e)(ii)	1 mark for ss showing inputs and output	1	
e.g. <pre> Enter a number 100 Enter a number 10 Enter a number 5 Enter a number 6 Enter a number 18 Enter a number 68 100 10 5 6 </pre>			

Question	Answer	Marks	Guidance
2(a)	1 mark each - Record structure / class with identifier <code>ToyRecord</code> - Four fields declared/initialised	2	Class needs a constructor for MP1 (allow/ignore parameters). Fields can be attributes declared by comment
Example program code Python <pre>class ToyRecord: def __init__(self, pName, pNumBricks, pDifficulty, pCategory): self.ToyName = pName #string self.NumberBricks = pNumBricks #integer self.Difficulty = pDifficulty #string self.Category = pCategory #string</pre> VB.NET <pre>Structure ToyRecord Dim ToyName As String Dim NumberBricks As Integer Dim Difficulty As String Dim Category As String End Structure</pre>			

Question	Answer	Marks	Guidance
	<pre>Java public class ToyRecord { public String ToyName; public Integer NumberBricks; public String Difficulty; public String Category; public ToyRecord(String pToyName, Integer pNumberBricks, String pDifficulty, String pCategory){ ToyName = pToyName; NumberBricks = pNumberBricks; Difficulty = pDifficulty; Category = pCategory; } }</pre>		

Question	Answer	Marks	Guidance
2(b)(i)	1 mark each • Function header taking parameter and returning populated array • Exception handling with suitable output (all file handling inside try) • Opening the file • ...closing the file in an appropriate place • Looping until EOF/16 times and reading in each line • Splitting data by comma (and removing new line) • Creating new <code>ToyRecord</code> with data // updating pre-existing record • Appending each record to array	8	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def ReadData(TheToys): try: TheFile = open("Toys.txt") for Line in TheFile: DataSplit = (Line.strip()).split(",") TheToys.append(ToyRecord(DataSplit[0],int(DataSplit[1]), DataSplit[2], DataSplit[3])) TheFile.close() except: print("Cannot open file") return TheToys</pre> VB.NET <pre>Function ReadData(TheToys() As ToyRecord) Dim TempData(3) As String Dim TempRecord As ToyRecord Try Dim FileReader As New System.IO.StreamReader("Toys.txt") While Not FileReader.EndOfStream TempData = (FileReader.ReadLine()).Split(",") TempRecord.ToyName = TempData(0) TempRecord.NumberBricks = TempData(1) TempRecord.Difficulty = TempData(2) TempRecord.Category = TempData(3) TheToys(NumberToys) = TempRecord NumberToys = NumberToys + 1 End While FileReader.Close() Catch ex As Exception Console.WriteLine("Cannot open or read from file") End Try Return TheToys End Function</pre>		

PUBLISHED

Question	Answer	Marks	Guidance
Java	<pre> public static ToyRecord[] ReadData(ToyRecord[] TheToys){ String[] TempData; try{ Scanner Scanner1 = new Scanner(new File("Toys.txt")); while(Scanner1.hasNextLine()){ TempData = Scanner1.nextLine().split(","); TheToys[NumberToys] = new ToyRecord(TempData[0],Integer.parseInt(TempData[1]),TempData[2],TempData[3]); NumberToys++; } Scanner1.close(); } catch (FileNotFoundException ex){ System.out.println("No file found"); } return TheToys; } </pre>		

Question	Answer	Marks	Guidance
2(b)(ii)	1 mark each • Procedure header, taking array parameter • Looping through each array element ... • ...outputting the data	3	
Example program code Python <pre>def OutputToys(TheToys): for Item in TheToys: print(Item.ToyName, Item.NumberBricks, Item.Difficulty, Item.Category)</pre> VB.NET <pre>Sub OutputToys(TheToys) For x = 0 To NumberToys - 1 Console.WriteLine(TheToys(x).ToyName & " " & TheToys(x).NumberBricks & " " & TheToys(x).Difficulty & " " & TheToys(x).Category) Next End Sub</pre> Java <pre>public static void OutputToys(ToyRecord[] TheToys){ for(Integer x = 0; x < NumberToys; x++){ System.out.println(TheToys[x].ToyName + " " + TheToys[x].NumberBricks + " " + TheToys[x].Difficulty + " " + TheToys[x].Category); } }</pre>			

Question	Answer	Marks	Guidance
2(b)(iii)	1 mark each • Declaring array of (16 <code>ToyRecord</code> elements) • Calling <code>ReadData()</code> with empty array parameter and storing return array • Calling <code>OutputToys()</code> with returned array	3	
Example program code Python <pre>TheToys = [] #16 elements TheToys = ReadData(TheToys) OutputToys(TheToys)</pre> VB.NET <pre>NumberToys = 0 Dim TheToys(15) As ToyRecord TheToys = ReadData(TheToys) OutputToys(TheToys)</pre> Java <pre>NumberToys = 0; ToyRecord[] TheToys = new ToyRecord[16]; TheToys = ReadData(TheToys); OutputToys(TheToys); OutputToys(TheToys)</pre> Java <pre>NumberToys = 0; ToyRecord[] TheToys = new ToyRecord[16]; TheToys = ReadData(TheToys); OutputToys(TheToys);</pre>			

Question	Answer	Marks	Guidance
2(b)(iv)	1 mark for screenshot showing the toy data output	1	Ignore format
e.g. <pre> Fire Station 3981 Expert Town Theatre 4120 Expert Town Riding Stable 1025 Medium Play Hospital 1220 Medium Play Super Car 235 Easy Play Vintage Car 352 Easy Play Tree House 5120 Expert Display Car Park 1124 Medium Town Flowers 1A 632 Medium Display Castle 2045 Medium Play Aeroplane large 2415 Medium Play Aeroplane medium 1520 Easy Play Motorbike large 1563 Medium Display Cafe 2041 Medium Town Shopping Mall 6512 Medium Town Park 1352 Easy Play </pre>			

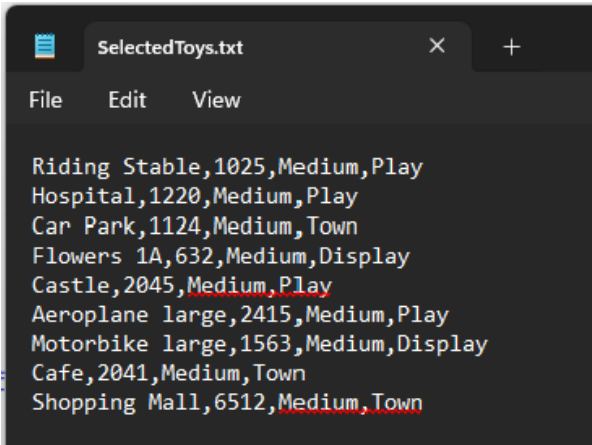
Question	Answer	Marks	Guidance
2(c)(i)	1 mark each - Function header taking array of <code>ToyRecord</code> and returning a populated array - Prompt to enter a difficulty level and reading as input repeatedly until valid - Iterating through each element in array and comparing difficulty to input appending those that match to the new array	5	Allow any input as long as equates to the 3 levels
Example program code Python <pre>def SearchToys(TheToys): SelectedToys = [] Difficulty = -1 while Difficulty < 1 or Difficulty > 3: Difficulty = int(input("Enter 1 for easy, 2 for medium and 3 for expert ")) if Difficulty == 1: Choice = "Easy" elif Difficulty == 2: Choice = "Medium" else: Choice = "Expert" for Item in TheToys: if Item.Difficulty == Choice: SelectedToys.append(Item) return SelectedToys</pre>			

Question	Answer	Marks	Guidance
VB.NET	<pre> Function SearchToys(TheToys) Dim SelectedToys(16) As ToyRecord Dim Difficulty As Integer = -1 Dim Choice As String NumberSelected = 0 While (Difficulty < 1 Or Difficulty > 3) Console.WriteLine("Enter 1 for easy, 2 for medium and 3 for expert ") Difficulty = Console.ReadLine() End While If Difficulty = 1 Then Choice = "Easy" ElseIf Difficulty = 2 Then Choice = "Medium" Else Choice = "Expert" End If For x = 0 To NumberToys - 1 If TheToys(x).Difficulty = Choice Then SelectedToys(NumberSelected) = TheToys(x) NumberSelected = NumberSelected + 1 End If Next Return SelectedToys End Function </pre>		

Question	Answer	Marks	Guidance
Java	<pre> public static ToyRecord[] SearchToys(ToyRecord[] TheToys){ ToyRecord[] SelectedToys = new ToyRecord[16]; Integer Difficulty = -1; String Choice = ""; NumberSelected = 0; Scanner Scanner1 = new Scanner(System.in); while(Difficulty < 1 Difficulty > 3){ System.out.println("Enter 1 for easy, 2 for medium and 3 for expert "); try{ Difficulty = Integer.parseInt(Scanner1.nextLine()); }catch(NumberFormatException ex){} } if(Difficulty == 1){ Choice = "Easy"; }else if(Difficulty == 2){ Choice = "Medium"; }else{ Choice = "Expert"; } for(Integer x = 0; x < NumberToys; x++){ if((TheToys[x].Difficulty).equals(Choice)){ SelectedToys[NumberSelected] = TheToys[x]; NumberSelected++; } } return SelectedToys; } </pre>		

Question	Answer	Marks	Guidance
2(c)(ii)	1 mark each to max 3 • Procedure header taking array of type <code>ToyRecord</code>, opening and closing text file • Looping through array and formatting the data correctly • ...writing each line to the file • Exception handling with suitable catch and output	3	
Example program code Python <pre>def WriteToys(TheToys): try: TheFile = open("SelectedToys.txt", 'w') for Item in TheToys: LineToWrite = Item.ToyName + "," + str(Item.NumberBricks) + "," + Item.Difficulty + "," + Item.Category + "\n" TheFile.write(LineToWrite) TheFile.close() except: print("Cannot write to file")</pre> VB.NET <pre>Sub WriteToys(TheToys) Dim TextToWrite As String = "" Try Dim FileWriter As New System.IO.StreamWriter("SelectedToys.txt") For x = 0 To NumberSelected - 1 TextToWrite = TheToys(x).ToyName & "," & TheToys(x).NumberBricks & "," & TheToys(x).Difficulty & "," & TheToys(x).Category FileWriter.WriteLine(TextToWrite) Next FileWriter.Close() Catch ex As Exception Console.WriteLine("Cannot open or read from file") End Try End Sub</pre>			

Question	Answer	Marks	Guidance
Java	<pre> public static void WriteToys(ToyRecord[] TheToys){ String TextToWrite = ""; File TheFile = new File("SelectedToys.txt"); try{ FileWriter FW = new FileWriter(TheFile, true); for(Integer x = 0; x < NumberSelected;x++){ TextToWrite = TheToys[x].ToyName + "," + TheToys[x].NumberBricks + "," + TheToys[x].Difficulty + "," + TheToys[x].Category + "\n"; FW.write(TextToWrite); } FW.close(); } catch(IOException ex){ System.out.println("File not found, cannot write."); } } </pre>		
2(c)(iii)	1 mark each • Calling SearchToys() with populated array and storing return array • Calling WriteToys() with returned array	2	
Example program code	Python <pre> SelectedToys = [] SelectedToys = SearchToys(TheToys) WriteToys(SelectedToys) </pre> VB.NET <pre> Dim SelectedToys(15) As ToyRecord SelectedToys = SearchToys(TheToys) WriteToys(SelectedToys) </pre> Java <pre> ToyRecord[] SelectedToys = new ToyRecord[16]; SelectedToys = SearchToys(TheToys); WriteToys(SelectedToys); </pre>		

Question	Answer	Marks	Guidance
2(c)(iv)	1 mark for screenshot of text file with correct name and content	1	Ignore format, check content
e.g.  <pre> Riding Stable,1025,Medium,Play Hospital,1220,Medium,Play Car Park,1124,Medium,Town Flowers 1A,632,Medium,Display Castle,2045,Medium,Play Aeroplane large,2415,Medium,Play Motorbike large,1563,Medium,Display Cafe,2041,Medium,Town Shopping Mall,6512,Medium,Town </pre>			

Question	Answer	Marks	Guidance
3(a)(i)	1 mark each • Class declaration (and end) with attributes • Constructor header (and end) • ... taking 2 parameters ... • ... assigning parameters to attributes	4	
Example program code Python <pre>class TheObject: def __init__(self, pType, pAmount): self.ObjectType = pType #string self.Amount = pAmount #integer</pre> VB.NET <pre>Class TheObject Public ObjectType As String Public Amount As Integer Sub New(pType, pAmount) ObjectType = pType Amount = pAmount End Sub End Class</pre> Java <pre>class TheObject{ public String Type; public Integer Amount; public TheObject(String pType, Integer pAmount){ ObjectType = pType; Amount = pAmount; } }</pre>			

Question	Answer	Marks	Guidance
3(a)(ii)	1 mark each • Class header (and end) and attributes including array • Constructor taking 1 parameter and storing in <code>NumberSpaces...</code> • ... initialising empty objects to 100 spaces in <code>BoardArray</code> • Get method with no parameter, returning attribute	4	
Example program code Python <pre> class Path: def __init__(self, pNumberSpaces): self.__BoardArray = [] #100 TheObject self.__NumberSpaces = pNumberSpaces #int for Number in range(100): self.__BoardArray.append(TheObject("", -1)) def GetNumberSpaces(self): return self.__NumberSpaces </pre> VB.NET <pre> Class Path Private BoardArray(99) As TheObject Private NumberSpaces As Integer Sub New(pNumberSpaces) For x = 0 To 99 BoardArray(x) = New TheObject("", -1) Next NumberSpaces = pNumberSpaces End Sub Function GetNumberSpaces() Return NumberSpaces End Function End Class </pre>			

Question	Answer	Marks	Guidance
Java	<pre>class Path{ private static TheObject[] BoardArray = new TheObject[100]; private static Integer NumberSpaces; public Path(Integer pNumberSpaces){ for(Integer x = 0; x < pNumberSpaces; x++){ BoardArray[x] = new TheObject("",-1); } NumberSpaces = pNumberSpaces; } public Integer GetNumberSpaces(){ return NumberSpaces; } }</pre>		

Question	Answer	Marks	Guidance
3(a)(iii)	1 mark each • Method header taking array as parameter • If more elements, returning <code>FALSE</code> • (otherwise) storing parameter elements in array and returning <code>TRUE</code>	3	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def InitialiseData(self, TheArray): ALength = len(TheArray) if ALength > self.__NumberSpaces: return False for x in range(len(TheArray)): self.__BoardArray[x] = TheArray[x] return True</pre> VB.NET <pre>Function InitialiseData(TheArray()) Dim ALength As Integer = TheArray.Length If ALength > NumberSpaces Then Return False End If For x = 0 To ALength - 1 BoardArray(x) = TheArray(x) Next Return True End Function</pre> Java <pre>public boolean InitialiseData(TheObject[] TheArray){ Integer ALength = TheArray.length; if(ALength > NumberSpaces){ return false; } for(Integer x = 0; x < NumberSpaces; x++){ BoardArray[x] = TheArray[x]; } return true; }</pre>		

Question	Answer	Marks	Guidance
3(a)(iv)	1 mark each - Method header and returning string - Creating string of all not null path objects type and amount	2	
Example program code Python <pre>def ReturnPath(self): ReturnString = "" for x in range(self.__NumberSpaces): TempString = self.__BoardArray[x].ObjectType + "-" + str(self.__BoardArray[x].Amount) + " " ReturnString = ReturnString + TempString return ReturnString</pre> VB.NET <pre>Function ReturnPath() Dim ReturnString As String = "" Dim TempString As String = "" For x = 0 To NumberSpaces - 1 TempString = BoardArray(x).ObjectType & "-" & (BoardArray(x).Amount).ToString + " " ReturnString = ReturnString + TempString Next Return ReturnString End Function</pre> Java <pre>public String ReturnPath(){ String ReturnString = ""; String TempString = ""; for(Integer x = 0; x < NumberSpaces; x++){ TempString = BoardArray[x].ObjectType + "-" + BoardArray[x].Amount + " "; ReturnString = ReturnString + TempString; } return ReturnString; }</pre>			

Question	Answer	Marks	Guidance
3(a)(v)	1 mark each Method header (and end) taking one parameter and returning array element at parameter	1	Accept returning the integer or integer –1. Solution does not need to check if valid index.
Example program code Python <pre>def GetPathObject(self, TheIndex): if(TheIndex >= NumberSpaces - 1): TheIndex = self.__NumberSpaces - 1 return self.__BoardArray[TheIndex]</pre> VB.NET <pre>Function GetPathObject(TheIndex) If (TheIndex >= NumberSpaces - 1) Then TheIndex = NumberSpaces - 1 End If Return BoardArray(TheIndex) End Function</pre> Java <pre>public TheObject GetPathObject(Integer TheIndex){ if(TheIndex >= NumberSpaces - 1){ TheIndex = NumberSpaces - 1; } return BoardArray[TheIndex]; }</pre>			

Question	Answer	Marks	Guidance
3(b)(i)	1 mark each • Creating 5 objects of type <code>TheObject</code> • ...with correct values ... • ... and storing in <code>ArrayOfObjects</code> • Creating new instance of <code>Path</code> with 5 elements • Calling <code>InitialiseData</code> for the path with <code>ArrayOfObjects</code> • Outputting the path using return path	6	

Question	Answer	Marks	Guidance
	Example program code Python <pre> ArrayOfObjects = [] ArrayOfObjects.append(TheObject("Money",50)) ArrayOfObjects.append(TheObject("Experience",100)) ArrayOfObjects.append(TheObject("Money",200)) ArrayOfObjects.append(TheObject("Money",20)) ArrayOfObjects.append(TheObject("Experience",30)) Path1 = Path(5) Path1.InitialiseData(ArrayOfObjects) print(Path1.ReturnPath()) </pre> VB.NET <pre> Dim ArrayOfObjects(4) As TheObject ArrayOfObjects(0) = New TheObject("Money", 50) ArrayOfObjects(1) = New TheObject("Experience", 100) ArrayOfObjects(2) = New TheObject("Money", 200) ArrayOfObjects(3) = New TheObject("Money", 20) ArrayOfObjects(4) = New TheObject("Experience", 30) Dim Path1 As Path = New Path(5) Path1.InitialiseData(ArrayOfObjects) Console.WriteLine(Path1.ReturnPath()) </pre> Java <pre> TheObject[] ArrayOfObjects = new TheObject[5]; ArrayOfObjects[0] = new TheObject("Money",50); ArrayOfObjects[1] = new TheObject("Experience",100); ArrayOfObjects[2] = new TheObject("Money",200); ArrayOfObjects[3] = new TheObject("Money",20); ArrayOfObjects[4] = new TheObject("Experience",30); Path Path1 = new Path(5); Path1.InitialiseData(ArrayOfObjects); System.out.println(Path1.ReturnPath()); </pre>		

PUBLISHED

Question	Answer	Marks	Guidance
3(b)(ii)	1 mark for screenshot showing the path	1	
e.g. Money-50 Experience-100 Money-200 Money-20 Experience-30			